Worst-case Bounds for Bin-Packing Heuristics with Applications to the Duality Gap of the One-dimensional Cutting Stock Problem

by

Hans J. H. Tuenter

A thesis submitted to the Faculty of Science
of The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

Research group of Management Mathematics
School of Mathematics and Statistics
The University of Birmingham, Edgbaston
Birmingham B15 2TT, England

December 1996

Summary

The thesis considers the one-dimensional cutting stock problem, the bin-packing problem, and their relationship. The duality gap of the former is investigated and a characterisation of a class of cutting stock problems with the next round-up property is given. It is shown that worst-case bounds for bin-packing heuristics can be and are best expressed in terms of the linear programming relaxation of the corresponding cutting stock problem. The concept of recurrency is introduced for a bin-packing heuristic, which allows a more natural derivation of a measure for the worst-case behaviour. The ideas are tested on some well known bin-packing heuristics and (slightly) tighter bounds for these are derived. These new bounds (in terms of the linear programming relaxation) are then used to make inferences about the duality gap of the cutting stock problem. In particular; these bounds allow à priori, problem-specific bounds. The thesis ends with conclusions and a number of suggestions to extend the analysis to higher dimensional problems.

"You must have a problem," ...

"Choose one definite objective and drive ahead toward it.

You may never reach your goal, but you will find something of interest on the way."†

To my son Peter.

†Quote attributed to Felix Klein [6, p. 419]

Acknowledgments

I would like to thank the following people who were involved in the completion of this project. My supervisor during this period: Prof. R. A. Cuninghame-Green, who had enough trust to give me a free reign to pursue my research interests, and Dr. S. Salhi, whose laugh and enthusiasm were a major encouragement. My thanks go out to those who provided me with the moral support, the peace of mind and a caring environment: my parents for having the faith, Lise for taking care of the most precious, and Ruby for the warmth and harmony she shared with me.

HANS TUENTER

The University of Birmingham
December 1996

Contents

List of Figures

List of Tables

Part I

Cutting Stock Problems

Chapter 1

Introduction

Whenever a larger unit of material needs to be divided into smaller units, at a minimal cost, we are dealing with a *Cutting Stock Problem (CSP)*. As an example consider the following problem. A steel manufacturer produces bars of a length of 10 meters. A client places an order for fifteen bars of a length of 4m and five of a length of 3m. How should the manufacturer cut his stock lengths to satisfy his client's demands at minimal cost? This type of problem occurs frequently in industry. Instead of metal one can take wood, cloth, paper, leather, et cetera. The scope can be widened to 2-dimensional problems, e.g. sheets of metal.

Whenever we have items which need to be packed into a larger unit, we are dealing with a packing problem. As an example consider the following. An airline is offered a large variety of cargo that needs to be shipped. There are parcels which weigh just a few kilos, there is machinery which weighs several thousands of kilos and a whole range of items with intermediate weights. With the total weight carried on an aeroplane being the limiting factor; how many are necessary to transport the goods? This type of problem occurs frequently in industry, in particular the cargo industry.

As a less obvious example consider a manufacturer who has hundreds of half finished goods that need a finishing operation on a machine. Some need just five minutes on the machine, others one hundred minutes and many need somewhere in this range. If the machines are operational for eight hours in a day; how many machines does he need to reserve to finish the job within one day?

Both these examples are instances of the *Bin-Packing Problem (BPP)*. In the former we have a bin with capacity equal to the maximum cargo weight an aeroplane can carry. In the latter we can interpret every machine as a bin with a capacity of 480 (min.) and the finishing times as items with a size ranging from 5 to 100.

The CSP and BPP are obviously related; if one has solved a cutting problem it immediately gives the solution to the problem of packing the items that have just been cut. It depends upon one's viewpoint and practical considerations whether or not a problem is best formulated and solved as a cutting or packing problem. Consider a jigsaw puzzle: to most people this is a packing problem, but to the manufacturer it constitutes a cutting problem. And, obviously, knowledge of how the board was cut into the puzzle pieces would solve the puzzler's packing problem. The same applies to the general problem: if one has a general method to solve the BPP one can use it to solve the CSP and vice versa.

1.1 Research Motivation

The original motivation to examine more closely the link between the CSP, the BPP and their respective solution methods, came from practical experiences in the carpet industry.[69] In this setting, although a company may receive many thousands of customer orders for room sized pieces a day, the orders are spread over the many different qualities and colours that are stocked. For most of the individual types of carpet this results in only a few orders a day, but for some, very popular lines this may result in a large number of orders. Although the objective in both cases is the same (trim-loss minimisation) the solution methods are quite different.

Following the classification, normally used in literature, the former set of problems would be classified as a BPP and the latter as a CSP.

	BPP	CSP
no. item types	many	few
no. items/no. item types	few	many

Table 1.1. Characteristics of the bin-packing problem vs. the cutting stock problem

A BPP typically consists of a large number of many different items, whereas a CSP consists of a large number of few different items. For a more detailed typology, see Dyckhoff and Finke.[20] Although, technically speaking, these are merely two different formulations of the same integer problem, the distinction is important for practical purposes. The BPP can be used to model a process of on-line sequential allocation, whereas a CSP represents an off-line allocation process. Furthermore, different solution procedures are used for each class.

A further motivation is given by the standard method of solving a CSP (see chapter 2.2). This method basically splits the problem into a 'large' component, which is solved optimally, and a 'smaller' residual component, which is formulated as a BPP.

A final motivation was given by the realisation that worst-case bounds for BPP-heuristics could be expressed in terms of the *Linear Programming (LP)* relaxation of the CSP.

Apart from these motivations the mere fact that the CSP and BPP are just two different interpretations/formulations of the same problem gives an intuitive notion that one should be able to use insights, results and techniques from one domain in the other.[1]

1.2 Notation & Terminology

In this section we introduce some notational conventions and terminology common to all chapters. It also serves as a quick overview of the main concepts. More specific notation, pertaining to the domains of cutting stock problems, and bin-packing heuristics can be found in chapters 2 and 5, respectively. For ease of reference most notation and acronyms are listed in a glossary (p. 261).

Special elements/captions such as a theorem, lemma, corollary and the like, are numbered consecutively within a chapter or appendix (e.g. lemma 3.14 and property B10). Equations are numbered, starting with (1), consecutively within a chapter. For example, (5.2) is the second equation in chapter 5. Numbers in square brackets refer to the equivalent publication in the list of references. Footnotes are indicated by a superscript and printed on the same page. Tables as well as diagrams are numbered consecutively within a chapter or appendix.

Occasionally a theorem, lemma or corollary is given a name and can be referred to by its name; viz. "by the integrality assumption (theorem A1), we have ...". These names are listed in the index.

[1] As Dyckhoff and Finke[20] note in the introduction to their book:

> "Because of the strong link between cutting and packing problems based on the duality of material and space, it seems obvious to examine both within a general framework. Statements made concerning cutting problems can also be significant for problems in packing, and vice versa."

A decision problem where the value of certain variables has to be determined in order to maximise (or minimise) an objective function will be referred to as a *program*. For instance the problem of deciding which vector out of a given set $\mathcal{A}$ maximises a function, translates into the program

$$\mathcal{D} = \boxed{\begin{array}{ll} \text{Max} & f(\mathbf{a}) \\ \text{st} & \mathbf{a} \in \mathcal{A} \end{array}} \tag{1.1}$$

We shall refer to $\mathcal{D}$ as the decision problem itself and frequently, if there is no confusion, as to its solution value. To highlight that the value of a decision variable is an optimal one, i.e. maximises (or minimises) the objective function, the star symbol is used. As in; '$\mathbf{a}^*$ maximises $\mathcal{D}$'. Sometimes, if we want to refer to the value of a (set of) particular feasible solution(s) with respect to a program, we will use $\mathcal{D}(\mathbf{a})$ or just $\mathcal{D}$, if there is no confusion. A variable is said to be *active*, with respect to a program, if it has a non-zero value. For instance; 'if a_1 is active then $\mathcal{D} < \mathcal{D}^*$, so that we may assume $a_1^* = 0$'. Decision problems, such as linear- or integer programs are treated as a separate entity. This is emphasised by packing them in 'boxes' as in (1.1).

A set of n items can be represented in different ways, depending on the domain that we are working in. The one that is mostly used for bin-packing is the list form and will be denoted as $\mathcal{L} = \{x_1, \ldots, x_n\}$, where x_i is the size of the i^{th} item. Usual convention dictates that the sizes are scaled, so that the bin has size one and all items have a size not exceeding one. We shall loosely use $\mathcal{L} \subset \langle 0, \varphi]$ to denote that all elements of the list $\mathcal{L}$ have a size in that interval.

In the setting of a cutting stock problem the order of the items does not matter and we can describe the item set in a more concise manner. The usual convention here is that the item sizes and the bin size are integers. The set is characterised by the *integer* parameters m, L and *integer* parameter vectors $\mathbf{d}$ and $\mathbf{f}$. This denotes that the set of items contains m different item-types with sizes $d_1, \ldots, d_m$ to be cut in frequencies $f_1, \ldots, f_m$ from the length L. If we want to stress that a list can be represented in this manner, by these parameters, we will use $\mathcal{L} \cong (m, L, \mathbf{d}, \mathbf{f})$.

The default assumption for a list is that it has sizes in the interval $\langle 0, 1]$. If we want to stress that sizes are taken from a subset of $\langle 0, 1]$, in particular that the size of the largest item does not exceed a certain (parametrised) value we will use the terminology parametric- or restricted list. For example; 'the parametric list $\mathcal{L}$ has sizes in $\langle 0, \varphi]$, where $\varphi \leqslant 1/2$'.

A *singleton bin* is a bin packed with exactly one item and the item it contains is referred to as a *singleton item.* If the last bin is a singleton bin, the item it contains is usually denoted as x or d_c and will be referred to as the *critical item.*

We interpret the various algorithms and heuristics as operators on a list of items, and the corresponding acronym will be used to denote this. For instance, packing a list $\mathcal{L}$ using the heuristic H will use a number of $H(\mathcal{L})$ bins. The minimal number of bins that is necessary to pack a list will be denoted as $OPT(\mathcal{L})$, where OPT can be interpreted as some algorithm that produces an optimal solution.[2]

Similar, CSP is an algorithm that returns the minimum number of stock lengths necessary to fulfil the demand. To distinguish between the value of CSP as an integer programming problem and its LP-relaxation we will use CSP_I and CSP_R respectively.

A quantity which is useful in the analysis of cutting and packing problems is the (scaled) amount of *material* to be cut or packed. In the context of a cutting stock problem it is defined as $Mat = \sum f_i d_i / L$, and in the context of a packing problem (where the sizes are already scaled) it is defined as $Mat = \sum x_i$.

In the notation we will use a minimality principle. If a quantity depends upon one or more parameters, and the value of some of these is not directly relevant for the ensuing section, or if it is obvious what the parameters are we shall omit them. For instance; the value of a cutting stock problem can be referred to as $CSP_I(\mathcal{L})$, $CSP_I(m, L, \mathbf{d}, \mathbf{f})$, $CSP_I(m)$, $CSP_I(\mathbf{d}, \mathbf{f})$, et cetera, depending on the feature we want to highlight.

The function $\alpha()$ is defined as $\alpha(d) \equiv \lfloor L/d \rfloor$ or $\alpha(x) \equiv \lfloor 1/x \rfloor$ in the setting of a CSP or BPP respectively. Sometimes, if there is no confusion, we will use α_i to denote $\alpha(d_i)$ or $\alpha(x_i)$. The symbol β is used in a similar fashion. Throughout this thesis α and β are *always* assumed to be integers.

A unit fraction is a rational of the form $1/i$, where i is a positive integer.

In the analysis we frequently use the notation x^+ to denote a number $x + \varepsilon$, with $\varepsilon > 0$ and sufficiently small.

[2]That $OPT(\mathcal{L})$ is calculable in a finite number of steps follows from the fact that, for a fixed list with a finite number of item types, the number of possibilities to pack a bin is finite, and one could (theoretically) determine the value of $OPT(\mathcal{L})$ by complete enumeration.

1.3 Readers' guide

An *elementary bound* that holds for any list $\mathcal{L}$ and any [bin-packing] heuristic H is given by

$$Mat(\mathcal{L}) \leqslant CSP_R(\mathcal{L}) \leqslant CSP_I(\mathcal{L}) = OPT(\mathcal{L}) \leqslant H(\mathcal{L}) \tag{1.2}$$

This bound links the main quantities studied and constitutes a good starting point to explain the overall structure of the thesis.

The first part of the thesis is concerned with the cutting stock problem and its LP-relaxation; $CSP_I(\mathcal{L})$ and $CSP_R(\mathcal{L})$. The second part deals with various bin-packing heuristics and their relation to the LP-relaxation of the associated cutting stock problem; $H(\mathcal{L})$ and $CSP_R(\mathcal{L})$. The third part combines results of the previous two and draws some conclusions on the use of bin-packing heuristics to derive (bounds for) the solution value of the cutting stock problem; $H(\mathcal{L})$ and $CSP_I(\mathcal{L})$.

To get a quick overview of the main ideas we recommend reading sections 2.3, 3.1, 5.3 and 5.4 and the following chapter contents listing.

1.4 Chapter contents listing

In this section we will give a concise overview of the topics studied in each chapter.

In **chapter 2** we give a formal description of the *one-dimensional cutting stock problem CSP* and describe the most commonly used heuristic (based upon linear programming) to solve it. Additionally some of the research questions regarding CSP are outlined.

In **chapter 3** we will study the *duality gap* of CSP, give simplified proofs for known instances with an important property known as the *Round Up (RU)* property, and determine a new class which possesses a related property known as the *Next Round-Up (NRU)* property.

In **chapter 4** we will study the *harmonic CSP*, the class to which most of the instances with the largest duality gaps found so far belong. It is shown, by an elementary analysis, that the harmonic CSP has the NRU-property. A tighter bound for, and an asymptotic characterisation of the duality gap of the harmonic CSP is derived.

In **chapter 5** we introduce the basic concept that allows a more structured analysis of *bin-packing heuristics*; recurrency. It is based upon the intuitive notion that the asymptotic ratio for a packing heuristic is determined by bin configurations that can occur multiple times in a heuristic packing; i.e. are recurrent. Furthermore, we will show that worst-case bounds for bin-packing heuristics can be and are best expressed in terms of the LP-relaxation of the associated cutting stock problem. Finally, a tentative solution approach to derive worst-case bounds in practice is given.

In chapters 6, 7 and 8 we will test the concept of recurrency on three well known heuristics. We stress that the main motivation to examine these heuristics is to provide a testbed for recurrency, and to show that the known bounds in terms of OPT are also valid in terms of CSP_R. That we arrive at slightly improved bounds in a number of cases is a fortunate byproduct.

In **chapter 6** we will analyse the *First-Fit (FF)* heuristic and derive a more general bound. This bound is a direct consequence of the properties of a recurrent weighting function for FF.

FF was studied principally to test and develop the ideas on recurrency. The asymptotic ratios are already known.[37] However, the recurrency concept allows a more elegant derivation of the worst-case bound. For restricted lists, i.e. $\mathcal{L} \subset \langle 0, 1/\alpha]$, $\alpha = \{2, 3, \ldots\}$ we improve upon the constant and derive tight worst-case bounds.

In **chapter 7** we will analyse the *Next-Fit Decreasing (NFD)* heuristic and show that the concept of recurrency leads very naturally to a weighting function. This in turn, almost immediately, leads to a knapsack problem on unit fractions, from which the asymptotic ratio follows directly. We strengthen the known bounds[4] by reducing the constant, both for the general case and for restricted lists.

In **chapter 8** we will analyse the *First-Fit Decreasing (FFD)* heuristic. Although recurrency leads to a weighting function in a fairly straightforward manner, the subsequent analysis is still a *very lengthy* one. We manage to strengthen the known bounds[3,37,38,72] for the general case. It is in the study of the parametric case that recurrence leads to a much simplified analysis and tighter worst-case bounds. Interestingly enough, it turns out that the parametric asymptotic ratio of FFD is connected to a subset-sum problem on unit fractions.

In the previous three chapters we have derived (stronger) worst-case bounds in terms of CSP_R. This gives bounds for the difference between the heuristic solution-value and CSP_R, and thus for the duality gap; $OPT - CSP_R$.

In **chapter 9** we apply the bounds derived for the various bin-packing heuristics to establish bounds for the duality gap of CSP. These bounds give some nice à priori characteristics of its magnitude. We then show how one can derive tighter worst-case bounds for specific instances for some of the heuristics studied. It is further shown that a reduction algorithm, developed in the analysis of FFD, is useful in its own right. It is actually capable of solving/reducing the size of some problems published in the literature. To illustrate the use of the concept of recurrency for higher dimensional problems, some results are derived for a square-packing heuristic.

In **chapter 10** we summarise with some conclusions and list some of the results achieved. The chapter is concluded with suggestions for further research.

Chapter 2

The one-dimensional cutting stock problem

2.1 Introduction

A supplier who stocks (metal) bars of a given, standard length receives an order from a customer for a set of lengths to be cut from his stock lengths. How should the supplier cut his stock in order to fulfil the customer's demand, whilst minimising the amount of stock used to do so? This problem is known as the one-dimensional Cutting Stock Problem (CSP).

The cutting stock problem made its first appearance in management literature in the 1950s.[21,53] It has wide applicability. Instead of metal, one can take wood, cloth, paper or any other type of material. Basically any entity that needs to be divided into smaller units at minimal cost constitutes a cutting stock problem. The scope can be widened to 2-dimensional problems, e.g. sheets of metal and 3-dimensional problems, e.g. container loading. Over the years a vast number of articles have been published in various fields of application. For a literature overview see Sweeney,[66] and Cheng et al.[9] Additionally Dyckhoff and Finke[20] contains a large number of references.

Every practical problem has its own specific requirements and limitations. As a result there is no one standard method to solve all instances of CSP. However, there is a generic model which serves as a general framework within which adaptations are made to suit the particular application. It is on this generic model that we will focus.

The generic *one-dimensional CSP* can be formulated as follows;

> "Given a stock length L and smaller lengths d_i, demanded in quantities f_i. What is the minimum number of stock lengths necessary to fulfil the demand?"

The corresponding mathematical model is

$$CSP_l = \boxed{\begin{array}{ll} \text{Min} & \sum x_j \\ \text{st} & \sum \mathbf{a}_j x_j \geqslant \mathbf{f} \\ & \mathbf{a}_j \in \mathcal{A} \text{ and } x_j \in \mathrm{N} \end{array}} \tag{2.1}$$

where the *pattern set* $\mathcal{A}$ is defined as

$$\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^m \mid \sum a_i d_i \leqslant L\} \tag{2.2}$$

This set represents all possible ways of cutting a length L into the smaller lengths $d_1, \ldots, d_m$. It can also be characterised by its *extremal patterns*;

$$\mathcal{E} = \{\mathbf{a} \in \mathcal{A} \mid \sum a_i d_i > L - d_{\min}\} \tag{2.3}$$

where $d_{\min}$ denotes the smallest size in the set $\{d_i\}$. The set $\mathcal{E}$ contains all patterns that have a wastage from which we cannot cut any more items.

The elements a_{ij} of pattern $\mathbf{a}_j$ give the number of items of length d_i that are generated by cutting one stock length according to pattern j. The requirement to fulfil all orders translates into $\sum_j a_{ij}x_j \geqslant f_i$ and gives the constraint in (2.1). The decision variables x_j denote the number of times a particular pattern is used. Since each pattern represents a stock length, the objective function $\sum x_j$ represents the number of stock lengths used.

Example The cutting stock problem is perhaps best illustrated by means of an example. Suppose that we stock lengths of size 10 and have 15 orders for the length 4, and 5 orders for the length 3. This would give $L = 10$, $\mathbf{d}^{\mathsf{T}} = [4,3]$ and $\mathbf{f}^{\mathsf{T}} = [15,5]$. The pattern sets $\mathcal{A}$ and $\mathcal{E}$ contain 9 and 3 patterns respectively.

$$\mathcal{A} = \left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \text{ and } \mathcal{E} = \left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix} \right\} \tag{2.4}$$

It should be clear that in solving (2.1) we can restrict ourselves to only extremal patterns. To illustrate this by the example; suppose that we have a solution with pattern $\mathbf{a}_4 = [1\ 1]$ active. We can replace this pattern by pattern $\mathbf{a}_3 = [1\ 2]$. This is still a feasible solution and does not increase the value of the objective function. Repeating this argument we see that one can discard the non-extremal patterns from the problem formulation. This gives the following integer program (IP) for the example.

$$CSP_I = \boxed{\begin{array}{ll} \text{Min} & x_1 + x_2 + x_3 \\ \text{st} & \begin{bmatrix}2\\0\end{bmatrix} x_1 + \begin{bmatrix}1\\2\end{bmatrix} x_2 + \begin{bmatrix}0\\3\end{bmatrix} x_3 \geqslant \begin{bmatrix}15\\5\end{bmatrix} \\ & x_{1,2,3} \in \mathbb{N} \end{array}} \tag{2.5}$$

It has a solution value of 9, using patterns [2 0] and [1 2] a total of 6 and 3 times respectively.

The approach of enumerating all [extremal] patterns, as we have tacitly suggested, and solving the corresponding IP was taken by several authors.[21, 53]

The catch however, is that for larger and even moderate sized problems this leads to IPs of enormous dimensions. The pattern set can easily number millions of patterns. Pierce[54] reports instances with over 1.5 million patterns, originating from the paper industry.

It was not until 1961, when Gilmore and Gomory[33] pioneered a delayed column-generation scheme in conjunction with linear programming, that it became feasible to solve CSPs on the scale encountered in practice.

Although this widened the scope of the CSP, there is still a (smaller) catch. Instead of solving CSP we actually solve its LP-relaxation; viz. CSP_R.

$$CSP_R : \boxed{\begin{array}{ll} \text{Min} & \sum x_j \\ \text{st} & \sum \mathbf{a}_j x_j \geqslant \mathbf{f} \\ & \mathbf{a}_j \in \mathcal{A} \text{ and } x_j \geqslant 0 \end{array}} \quad \text{dual :} \boxed{\begin{array}{ll} \text{Max} & \sum f_i u_i \\ \text{st} & \langle \mathbf{a}, \mathbf{u} \rangle \leqslant 1,\ \forall\, \mathbf{a} \in \mathcal{A} \\ & \mathbf{u} \geqslant \mathbf{0} \end{array}} \tag{2.6}$$

Relaxing the integer constraint $x_j \in \mathbb{N}$ in (2.1) to $x_j \geqslant 0$ in order to arrive at a linear program allows one to use standard LP-methods. However, it does not (directly) solve the original problem.

Before returning to this we will discuss the solution procedure for CSP_R in a bit more detail. CSP_R can be solved as any LP-problem, with the only difference being how one determines the column that enters the basis. This is done by a process called delayed *column-generation.*

If $\mathbf{u}_k$ is the dual multiplier associated with the basis in the k^{th} iteration, then this basis is optimal if $\langle \mathbf{a}, \mathbf{u}_k \rangle \leqslant 1$ holds for all $\mathbf{a} \in \mathcal{A}$. Otherwise we select the pattern $\mathbf{a}$ that is 'most violating', that is maximises $\langle \mathbf{a}, \mathbf{u_k} \rangle$. This means that the column to enter the basis can be generated from the following problem, which is known as the *unbounded knapsack problem (UKP)*, see for instance [49, pp. 91–103].

$$\boxed{\begin{array}{ll} \text{Max} & \sum a_i u_i \\ \text{st} & \mathbf{a}_j \in \mathcal{A} \end{array}} = \boxed{\begin{array}{ll} \text{Max} & \sum a_i u_i \\ \text{st} & \sum a_i d_i \leqslant L \\ & a_i \in \mathbb{N} \end{array}} \tag{2.7}$$

More information on linear programming in general can be found in [67, pp. 2–37]. A detailed example, illustrating the use of delayed column-generation in the cutting stock problem, can be found in [10, pp. 195–212].

At this point it is convenient to introduce some notation, terminology and normalising assumptions for the cutting stock problem.

We define α_i as $\alpha_i \equiv \lfloor L/d_i \rfloor$. The *wastage* of a pattern $\mathbf{a}_j$ is defined as $w_j \equiv L - \sum a_{ij} d_i$. Similarly, the scaled wastage in the IP- and LP-solution are defined as

$$\begin{aligned} W_I &= CSP_I - Mat \\ W_R &= CSP_R - Mat \end{aligned} \tag{2.8}$$

A *cut to destruct (CTD)* pattern[1] is defined as a pattern with zero wastage.

2.1 Assumption $L \geqslant d_1 > \cdots > d_m > 0$

2.2 Assumption $\gcd\{d_i\} = 1$

The first assumption is a matter of grouping. For the second we note that although $(L, \mathbf{d})$ are defining elements, it is the resulting pattern set $\mathcal{A}$, given by (2.2), that defines CSP. It is easily seen that, if $d = \gcd\{d_i\} > 1$, we can scale the generating equation of $\mathcal{A}$ by taking out the common factor d. So the transformation $d_i := d_i/d$ and $L := (L \text{ div } d)$, where div denotes the integer division, produces an equivalent problem. This illustrates that the pair $(L, \mathbf{d})$ that defines a particular pattern-set is not unique (in this context see also appendix B.7).

In the following lemmas and corollary we use $\mathbf{x}^*$ to denote an optimal solution to $CSP_R(\mathbf{f})$ and $\mathbf{u}^*$ to denote an optimal solution to its dual.

[1] The term originates from the carpet industry.[69]

2.3 Lemma *If* $x_j^* \geqslant 1$ *then* $CSP_R(\mathbf{f}) = 1 + CSP_R(\mathbf{f} - \mathbf{a}_j)$

Proof. *For brevity we use* $(\mathbf{f})$ *to denote the cutting stock problem defined on* $\mathbf{f}$. *Since* $\mathbf{x}^*$ *is an optimal solution to* $(\mathbf{f})$ *and* $x_j^* \geqslant 1$, *it follows that* $\mathbf{x}^* - \mathbf{e}_j$ *is a feasible solution to* $(\mathbf{f} - \mathbf{a}_j)$, *and thus* $CSP_R(\mathbf{f} - \mathbf{a}_j) \leqslant CSP_R(\mathbf{f}) - 1$. *If* $\mathbf{y}^*$ *is an optimal solution to* $(\mathbf{f} - \mathbf{a}_j)$ *then* $\mathbf{y}^* + \mathbf{e}_j$ *is a feasible solution to* $(\mathbf{f})$ *and thus* $CSP_R(\mathbf{f}) \leqslant CSP_R(\mathbf{f} - \mathbf{a}_j) + 1$. *Combining the two bounds proves the lemma.* □

2.4 Lemma $\forall i \leqslant j \;\; u_i^* \geqslant \lfloor d_i/d_j \rfloor u_j^*$

Proof. *For every* i *there is an* $\mathbf{a} \geqslant \mathbf{e}_i$ *with* $\langle \mathbf{a}, \mathbf{u}^* \rangle = 1$, *otherwise we can increase* u_i^*. *If* $\mathbf{a}$ *is such a pattern, then* $\mathbf{a} - \mathbf{e}_i + \lfloor d_i/d_j \rfloor \mathbf{e}_j$, $(i \leqslant j)$ *is a feasible pattern so that* $\langle \mathbf{a} - \mathbf{e_i} + \lfloor d_i/d_j \rfloor \mathbf{e_j}, \mathbf{u}^* \rangle \leqslant 1$ *holds. And therefore* $u_i^* \geqslant \lfloor d_i/d_j \rfloor u_j^*$, *from which the lemma follows.* □

2.5 Corollary $u_1^* \geqslant \cdots \geqslant u_m^* \geqslant 0$

2.6 Lemma $u_i^* \leqslant 1/\alpha_i$

Proof. $\alpha_i \mathbf{e}_i$ *is a feasible pattern, so that* $\alpha_i u_i^* \leqslant 1$ *must hold, and the lemma follows.* □

2.7 Lemma *All (active) patterns in an optimal solution of* CSP_R *satisfy* $\sum a_i/\alpha_i \geqslant 1$

Proof. *Suppose a pattern with* $\sum a_i/\alpha_i < 1$ *is used* $x > 0$ *times. Replace this pattern by patterns* $\alpha_i \mathbf{e}_i$ *used* a_i/α_i *times,* $1 \leqslant i \leqslant m$. *This reduces the value of the objective function by an amount of* $(1 - \sum a_i/\alpha_i) \times x$, *so that this pattern could not have been active in an optimal solution.* □

2.2 Standard solution method

A solution method commonly used for the one-dimensional cutting stock problem can be described as follows. Solve the LP-relaxation CSP_R, by means of the simplex method, and round up the value of the variables in the optimal solution to arrive at a feasible solution to CSP_I.

In practice, where one is usually dealing with high volumes, this works acceptably well. This is due to the fact that the optimal LP-solution, when using the simplex method, has at most m, the row-dimension of (2.1), variables active. This gives a heuristic solution for which the following holds.

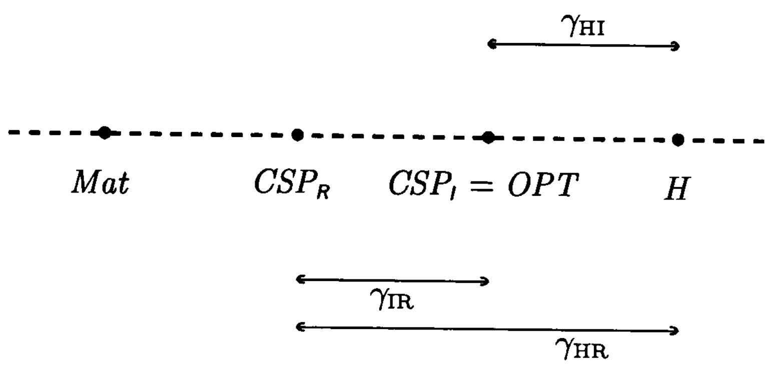
$$CSP_H - CSP_I \leqslant CSP_H - CSP_R = \sum \lceil x_j^* \rceil - \sum x_j^* < m \tag{2.9}$$

So that for high volume problems, i.e. $CSP_I \to \infty$ the relative error is negligible.

There are many variations on this scheme, but all rely on some form of rounding of the optimal LP-solution.

2.3 Research Questions regarding CSP

The idea of the dissertation is perhaps best illustrated by the following diagram, which is basically a visualisation of the elementary bound (1.2).



$$\begin{aligned} \gamma_{\mathrm{HI}} &= H - CSP_I \\ \gamma_{\mathrm{IR}} &= OPT - CSP_R \\ \gamma_{\mathrm{HR}} &= H - CSP_R \end{aligned}$$

Diagram 2.1. Visualisation of the elementary bound

The objective of a Cutting Stock Problem is to solve CSP_I. All the other elements; the LP-relaxation, its solution and the ensuing heuristic are only means to an end. Since BPP (and thus CSP) is NP-complete, see [31, p. 226], one has to resort to a heuristic to derive a good solution. In this context one is naturally interested in γ_{HI}, that is how close is the heuristic solution-value to the optimum solution-value. Inherently, this is a difficult question, and rather than investigating γ_{HI} directly, we will investigate γ_{IR} and γ_{HR}. This is done in chapters 3 and 4, and chapters 5–8 respectively.

There are some important questions that are suggested by diagram 2.1. We do not manage to completely resolve these questions, but they have been a major motivation in our investigations.

1. What is the maximum difference between CSP_I and CSP_R?

   a) Can this difference be upper bounded by a constant, independent of the problem parameters?

   b) Failing this, can one derive bounds based on some characterisation of the problem instance, in particular in terms of the parameters $m, L, \mathbf{d}, \mathbf{f}$?

2. What is a good heuristic to convert an LP-solution into an IP-solution?

Chapter 3

Duality gap of CSP

3.1 Introduction and practical observations

The duality gap $\gamma(\mathcal{L})$ for an instance of CSP, defined by a list $\mathcal{L}$, is defined as the difference between the value of the integer program and its LP-relaxation.

3.1 Definition $\gamma(\mathcal{L}) = CSP_I(\mathcal{L}) - CSP_R(\mathcal{L})$

The duality gap for [the problem class of] CSP is defined as the maximum of $\gamma(\mathcal{L})$, taken over all problem instances.

3.2 Definition $\gamma = \max_{\mathcal{L}} \gamma(\mathcal{L})$

The definition of the duality gap allows us to define the following properties. If for a particular list we have $\gamma(\mathcal{L}) < 1$ we say that it [and the corresponding cutting stock problem] has the *round-up (RU)* property. The practical importance of instances having the RU-property is that the value of CSP is the rounded-up LP-value. That is $CSP_I(\mathcal{L}) = \lceil CSP_R(\mathcal{L}) \rceil$ and thus that solving a linear program solves CSP. Similarly, a list is said to have the *next round-up (NRU)* property if $\gamma(\mathcal{L}) < 2$. This implies that $CSP_I(\mathcal{L}) = \{0,1\} + \lceil CSP_R(\mathcal{L}) \rceil$.

An observation that was made by various practitioners, viz. Diegel[19] and Städtler,[65] was that (almost) all instances of CSP have the RU-property. Numerical tests on randomly generated data sets enforced this belief. As a result it was tacitly assumed that the RU-property is valid for CSP, or at least that for practical purposes this assumption could safely be made. Diegel[19] based an algorithm, which was used for problems in the paper industry, on this assumption. Städtler[65]

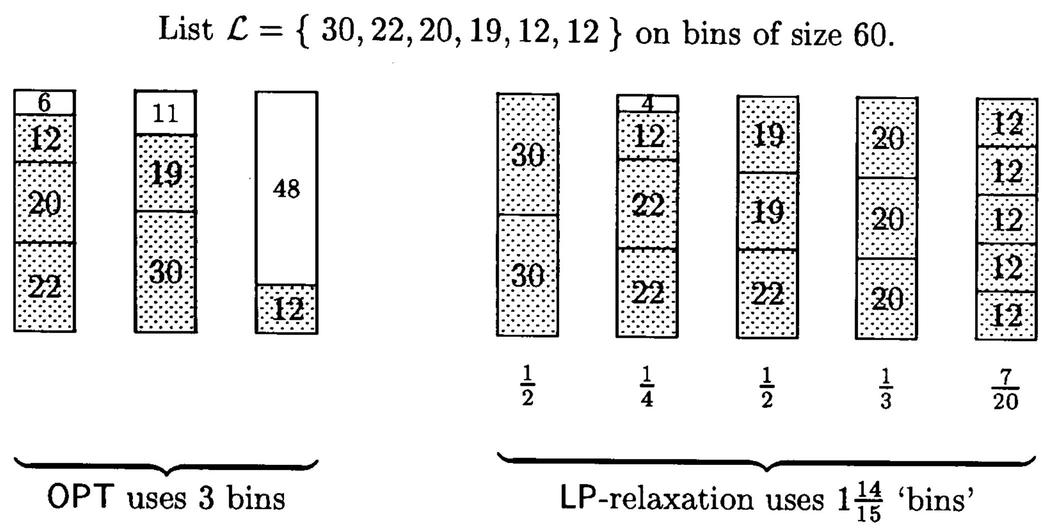


Diagram 3.1. Instance of CSP with $\gamma(\mathcal{L}) = 16/15$.

reports on problems in the aluminium industry, which all turned out to have the RU-property, and mentions that most of the heuristic solutions could be proved optimal by comparison with the value of the LP-relaxation.[1]

It was not until 1986 that Marcotte[48] showed that to determine whether or not an instance of CSP has the RU-property is NP-hard. Moreover, she constructed an instance with a gap of exactly 1. This example was based on extremely large numbers, which one would rarely encounter in practice. This still made it plausible that the RU-property would hold in practice. However, in 1990 an example with a gap of $1\frac{1}{30}$ was presented by Fieldhouse.[24] This example was based on numbers small enough to occur in practice. In a further paper[25] he identified several instances with a gap > 1 in a family of problems. In 1991, Scheithauer and Terno[60] found a member of this family with a gap of $1\frac{5}{132}$. Both these instances are replicated in table 3.1 (p. 24). Finally, in 1994 the largest known gap (up to date) was found by Gau[32] who gave an instance with a gap of $1\frac{1}{15}$. This instance consists of sizes of length $5000, 3750, 3250, 3001$ and 2000 to be cut from a length of 10000 in quantities $1, 1, 1, 1$ and 2 respectively.[2]

An observation[3] that is useful to eliminate some [trivial] cases is

$$Mat \leqslant 1 \Leftrightarrow CSP_R \leqslant 1 \Leftrightarrow CSP_I = 1 \tag{3.1}$$

so that in the search for instances with $\gamma \geqslant 1$ we may assume that $Mat > 1$ and thus $OPT \geqslant 2$.

[1]To be more specific, if $H(\mathcal{L})$ is the value of the heuristic solution for the instance $\mathcal{L}$, then $H(\mathcal{L}) = \lceil CSP_R(\mathcal{L}) \rceil$ held for these instances.

[2]Using the procedure outlined in appendix B.7 we can find an equivalent instance, which is given in diagram 3.1.

[3]We note that (3.1) does *not* hold in general for higher dimensional problems, see page 157.

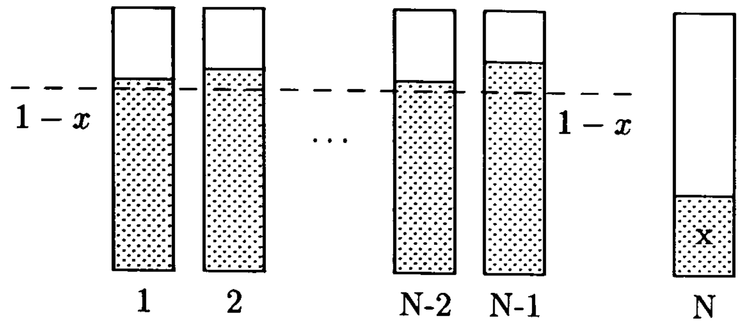


Diagram 3.2. Singleton-bin configuration.
The smallest item, x is packed in the last bin as a singleton item. All other bins are filled to a level strictly greater than $(1-x)$.

3.2 Canonical form of maximum gap instances

There are some assumptions that can be made on the instances that maximise the duality gap. These assumptions correspond to the two defining elements of γ; the optimal bin-configuration it packs into and the LP-relaxation.

3.3 Assumption *(Singleton-bin configuration)* *We may assume that the list that maximises $\gamma(\mathcal{L})$ packs into a configuration with the smallest item (in the list) packed into a singleton bin.*

Proof. *Suppose an optimal packing of $\mathcal{L}$ uses $N = OPT(\mathcal{L})$ bins. Now keep deleting the smallest item from the list until the resulting list $\mathcal{L}_0$ can be packed in $N-1$ bins. If x is the last item deleted, then the list $\mathcal{L}_0 + x$ can be packed into N (but no fewer) bins, with x as a singleton. Since CSP_R is monotonic it follows that $\gamma(\mathcal{L}) \leqslant \gamma(\mathcal{L}_0 + x)$ which proves the assumption.* □

In the singleton-bin configuration all bins, except the singleton bin, are filled to a level strictly greater than $1-x$, and there are $OPT-1$ such bins. For $OPT \geqslant 2$, the total amount of material to be packed satisfies $Mat > (OPT-1)(1-x)+x$. This gives rise to the following bound, which we will sometimes refer to as the *naïve bound*, for the list that maximises $\gamma(\mathcal{L})$.

$$OPT \geqslant 2 \;\Rightarrow\; OPT < 2 + \frac{Mat-1}{1-x} \tag{3.2}$$

3.4 Assumption *(Residual CSP)* *We may assume that the list that maximises $\gamma(\mathcal{L})$ is a residual CSP; that is all variables in the optimal LP-solution to CSP_R have a value strictly less than one.*

Proof. *Let $\mathbf{x}^*$ be an optimal solution to $CSP_R(\mathbf{f})$ and suppose that there is an $x_j^* \geqslant 1$. Now use $CSP_I(\mathbf{f}) \leqslant 1 + CSP_I(\mathbf{f}-\mathbf{a}_j)$ and lemma 2.3 to give $\gamma(\mathbf{f}) \leqslant \gamma(\mathbf{f}-\mathbf{a}_j)$. Repeating the argument we see that $\gamma(\mathbf{f}) \leqslant \gamma(\mathbf{r})$ holds, where the vector of residual frequencies is defined as $\mathbf{r} = \sum \mathbf{a}_j(x_j^* \bmod 1)$.* □

There are many other minimality assumptions one can make, but these are not so easily incorporated in the analysis of the duality gap. For instance,

- Cutting any item in two (or more) items must 'collapse' the bin configuration; i.e. the resulting list can be packed into one less bin. This applies in particular to the singleton item x.
- If we scale the list to integers, so that the bin size is L and the critical item has size d_c, then there must be at least one bin filled to a level $L - d_c + 1$.

However, for a given list these minimality assumptions sometimes allow us to construct a list with a (slightly) larger gap (see chapter 8).

The above assumptions imply that in the search for a maximum-gap CSP we can restrict ourselves to instances of the residual CSP, which pack into a singleton-bin configuration.

3.3 Bounds on the duality gap

In this section we will derive some bounds for the duality gap, which are valid for all lists. First, we list the following lemmas, these are known results but easily derived.

3.5 Lemma *For any positive integer α and list $\mathcal{L} \subset \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ the bound $\gamma(\mathcal{L}) < 1$ holds.*

Proof. *If there are n items in the list then $CSP_R = n/\alpha$ and $CSP_I = \lceil n/\alpha \rceil$. This gives a bound for the duality gap as $\gamma \leqslant 1 - 1/\alpha$, and proves the lemma.* □

By the theory of linear programming, we know that there is a solution with at most m (the row dimension of the LP) variables active. Constructing a heuristic solution by rounding up the optimal LP-solution gives the bound in the following corollary.

3.6 Corollary $\gamma(\mathcal{L}) < m$

The same bound is given by Coffman and Lueker [15, p. 32], with the slight difference that their bound contains a 'less than or equal to' sign. Note that both the lemma and corollary prove that any instance of CSP with only one item type has the RU-property.

3.7 Lemma *Any instance of CSP with at most two different item sizes has the RU-property.*

Proof. *It suffices to show that the lemma holds for a residual CSP, since this has a duality gap which is not larger than the CSP from which it is derived. If the residual CSP has $Mat \leqslant 1$*

then (3.1) implies $\gamma < 1$. *If Mat* > 1 *then (3.1) implies* $CSP_R > 1$ *and since there exists a solution [to the residual CSP] with* $OPT \leqslant m = 2$, *this also gives* $\gamma < 1$, *and proves the lemma.* □

We now derive a new lemma, which sharpens corollary 3.6 and lemma 3.7. To this end we use a simple characterisation of γ, which can be derived from (2.8); $\gamma = W_I - W_R$. In this, W_I and W_R are the wastage in the optimal packing of $\mathcal{L}$ and the fractional packing given by the LP-solution, respectively. Combining this with the singleton-bin configuration (diagram 3.2) we get an upper bound for γ as follows, where x is the size of the critical item.

$$\gamma = W_I - W_R \leqslant W_I < (OPT - 1)x + 1 - x = 1 + (OPT - 2)x, \quad \text{for } OPT \geqslant 2 \tag{3.3}$$

But we can do better: by the same rationale, any residual CSP has an optimal integer solution which is less than or equal to m. Since the residual CSP has a row dimension which is no larger than the row dimension of the original CSP, we can combine this with (3.3) to the following bound.

$$m \geqslant 2 \;\Rightarrow\; \gamma(\mathcal{L}) < 1 + (m - 2)x \tag{3.4}$$

We now use this bound in the following lemma.

3.8 Lemma *If the largest item in a list,* $\varphi \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ *then* $\gamma(\mathcal{L}) < 1 + \frac{m-2}{\alpha+1}$
Proof. *For* $m \geqslant 2$ *and* $x \leqslant \frac{1}{\alpha+1}$ *it follows from (3.4). Otherwise, it follows from lemma 3.5.* □

3.9 Corollary *Any instance of CSP with* m *item-types, largest item* $\varphi \leqslant 1/\alpha$, *and* $m \leqslant \alpha + 3$ *has the NRU-property.*

3.4 CSPs with the Round-Up property

Marcotte[47] proves that certain classes of CSP have the round-up property. Her proofs are based on the concept of the 'integral decomposition property' of the associated knapsack polyhedron, see Baum and Trotter.[5] We give new and simpler proofs for these classes, which do not make use of this concept. We reduce these instances to CSPs for which the RU-property has already been established, viz. those with only one or two item types.

We assume that all sizes d_i are integers and that L is the stock length. An instance of CSP is said to be *successive divisible* if $d_m | d_{m-1} | \cdots | d_1$.

3.10 Lemma *If the smallest item in the list* $\mathcal{L}$ *is an item with size 1, then one of the following holds.*

(1) *$\mathcal{L}$ has the RU-property; $\gamma(\mathcal{L}) < 1$.*

(2) *Removing all items with size 1 creates a list $\mathcal{L}'$ with $\gamma(\mathcal{L}') \geqslant \gamma(\mathcal{L})$.*

Proof. *Assume we have a list $\mathcal{L}$ with f items of size 1 and let $\mathcal{L}'$ be the list with all items 1 removed. Denote by W_I and W_I' the wastage in the $CSP_I(\mathcal{L})$ and $CSP_I(\mathcal{L}')$-solution respectively. We have the following two cases;*

(1) *$f > W_I' \times L$: Take the solution to $CSP_I(\mathcal{L}')$ and pack the $W_I' \times L$ items of size 1 in its bins. The remainder of the items (of size 1) can be packed into $\lceil f/L - W_I' \rceil$ bins. This creates a solution to $CSP_I(\mathcal{L})$ with bins which are all CTDs, except possibly one. Therefore $W_I \leqslant (L-1)/L$ and thus $\gamma(\mathcal{L}) \leqslant W_I < 1$.*

(2) *$f \leqslant W_I' \times L$: All the f items of size 1 will fit in the (wastage of the) bins of the $CSP_I(\mathcal{L}')$ solution, so that $CSP_I(\mathcal{L}') = CSP_I(\mathcal{L})$. Since CSP_R is monotonic it follows that $CSP_R(\mathcal{L}) \geqslant CSP_R(\mathcal{L}')$ and thus $\gamma(\mathcal{L}') \geqslant \gamma(\mathcal{L})$.*

This proves the two cases of the lemma. □

3.11 Lemma *The successive divisible CSP has the RU-property.*

Proof. *By contradiction. Assume that we have a list $\mathcal{L}$, where all the sizes of the item types are successive divisible and $\gamma(\mathcal{L}) \geqslant 1$. By assumption 2.2, we may assume that the smallest item, $d_m = 1$. But then the second case of lemma 3.10 must hold. Now remove all the items of size 1 to create a list $\mathcal{L}'$ with $\gamma(\mathcal{L}') \geqslant \gamma(\mathcal{L})$. The list $\mathcal{L}'$ inherits the successive divisibility from $\mathcal{L}$ and we can apply the same rationale again. This creates a sequence of lists each with one less item type and a larger gap than its predecessor. Furthermore, the last list in the sequence has only one item type. But, by lemma 3.7, this last instance has the RU-property and therefore all its predecessors have, which includes the original list. The assumption was therefore false and thus proves the lemma.* □

The proof of lemma 3.11 contains the seeds for an algorithm to construct a solution to the successive divisible CSP. First pack the items of largest size in a minimal number of bins. Then pack as many items of the next largest size in the waste of the current bins and pack the remainder in new bins. Continue doing this until all items are packed. This happens to be the well known FFD-heuristic (see chapter 8). That FFD is optimal for instances of CSP which have the property of successive divisibility is also noted by Marcotte,[47] and Coffman et al.[13] An elementary proof is given in the following lemma.

3.12 Lemma *FFD is optimal for the successive divisible CSP.*

Proof. *Suppose FFD uses N bins and the largest (first) item in the last bin has size d_c. Now delete all items from the list that come after d_c to leave a singleton-bin configuration (see diagram 3.2). We can scale this new list so that the smallest item (in bin N) has size 1. But this means that the first $N-1$ bins are all CTDs, so that $L \times Mat = (N-1)L + 1$ from which $N < 1 + Mat$ follows. Since $Mat \leqslant OPT \leqslant FFD$ and the latter two are integer, it follows that FFD is optimal.* □

3.13 Lemma $d_m | \cdots | d_{p+1}$, $d_p | \cdots | d_1$ *where* $d_{p+1} \not| \, d_p$ *and* $\forall i \; d_i | L \Rightarrow$ *CSP has the RU-property.*

Proof. *It is not difficult to see that $CSP_R = \sum f_i d_i / L = Mat$. Define $\alpha_i = L/d_i$ and set $d_0 = L$. If there is an $i \neq 1, p+1$ such that $f_i \geqslant d_{i-1}/d_i$ then we can replace (d_{i-1}/d_i) items of size d_i by one item of size d_{i-1}. This does not alter CSP_R but may increase CSP_I and therefore may increase the gap. Similarly for $i = 1, p+1$; if $f_i \geqslant \alpha_i$ we can replace α_i items of size d_i by one item of size 1. This gives rise to the following assumptions*

$$f_i \leqslant \frac{d_{i-1}}{d_i} - 1, \quad \text{for } i \neq 1, p+1 \qquad \text{and} \qquad f_i \leqslant \alpha_i - 1, \quad \text{for } i = 1, p+1 \tag{3.5}$$

The amount of material to pack is given by

$$Mat_1 = \sum_{i=1}^{p} f_i d_i / L \leqslant (\alpha_1 - 1) d_1 / L + \sum_{i=2}^{p} (d_{i-1}/d_i - 1) d_i / L = 1 - d_p / L < 1 \tag{3.6}$$

and $Mat_2 = \sum_{i=p+1}^{m} f_i d_i / L \leqslant 1 - d_m / L < 1$. Now condition on the total amount of material to be packed:

(i) $Mat_1 + Mat_2 \leqslant 1$, then all the items will fit in one bin and thus $\gamma < 1$

(ii) $Mat_1 + Mat_2 > 1$, then we can fit all items $d_1, \ldots, d_p$ in one bin and all items $d_{p+1}, \ldots, d_m$ in a second bin. This gives $\gamma \leqslant 2 - (Mat_1 + Mat_2) < 1$

So, the maximum gap over CSPs of the above form is smaller than 1. □

3.5 CSPs with the Next Round-Up property

We can use bound (3.4) to identify a class of problems which have the NRU-property. Assume that we have item types $d_1 > \cdots > d_m$, and keep deleting the smallest item until we end up with a singleton-bin configuration, as in assumption 3.3. Now suppose that the critical item is an item with size d_i. This means that we can formulate an instance of CSP on the item types $d_1, \ldots, d_i$

with a gap at least as large. So that $\gamma < 1 + (i-2) \times d_i/L$ for $i \geqslant 3$, and $\gamma < 1$ for $i = 1, 2$. This immediately gives the following bound and the subsequent corollary, where $\alpha_i \equiv (L \text{ div } d_i)$.

$$\gamma(L, \mathbf{d}) < 1 + \max_{i \geqslant 3} \left\{ (i-2)\tfrac{d_i}{L} \right\} \tag{3.7}$$

3.14 Corollary *Any instance of CSP that satisfies* $\forall i \;\; i \leqslant 2 + \alpha_i$ *has the NRU-property.*

Note that this corollary represents a generalisation of corollary 3.9. For $m = 1$ this is obvious, and for $i \geqslant 2$ or $i \leqslant m - 1$ we have that $m \leqslant \alpha_1 + 3$ implies $i \leqslant \alpha_i + 2$.

3.6 Gap-increasing Transformations

In this section we will give some transformations which will transform an instance of CSP into one with a gap at least as large. This gives rise to certain assumptions on the instances that achieve a maximum gap. Additionally, this will lead to a natural derivation of the class of problems that contains the maximum-gap examples of Fieldhouse, and Terno and Scheithauer.

In the following, let $\mathcal{L}^*$ denote a list with maximal gap. Since we already know that $\gamma(\mathcal{L}^*) > 1$ we can make the following assumption by lemma 3.10.

3.15 Assumption $\mathcal{L}^*$ *does not contain any items of size* 1.

3.16 Assumption *If* $\mathcal{L}^* \cong (m, L, \mathbf{d})$ *then* $\forall i \;\; (L \text{ div } d_i) + (L \bmod d_i) \leqslant d_i - 1$

Proof. *Suppose there is an* i *such that the assumption does not hold. For notational convenience denote* $d = d_i$, $\alpha = (L \text{ div } d_i)$ *and* $\beta = (L \bmod d_i)$, *so that the assumption becomes* $\alpha + \beta \geqslant d$.

- *First we show that* $L \geqslant \alpha + 2$ *holds. For* $d \geqslant 3$ *we have* $L = \alpha d + \beta \geqslant 3\alpha \geqslant \alpha + 2$, *since* $\alpha \geqslant 1$. *For* $d = 2$ *and* $\alpha + \beta \geqslant 2$ *we have* $L = 2\alpha + \beta \geqslant \alpha + 2$. *The case* $d = 2$ *and* $\alpha + \beta = 1$ *cannot occur, since* $\alpha + \beta \geqslant d$ *by assumption. The case* $d = 1$ *cannot occur by assumption 3.15.*
- *We now construct a new instance by replacing one item of size* d *by an item of size* $d' = \frac{L}{\alpha+1}$, *so that* $d - 1 < d' < d$ *holds. This implies* $CSP_I' = CSP_I$ *and since we have reduced an item in size, that* $CSP_R' \leqslant CSP_R$. *The new CSP has therefore a gap at least as large. Scaling this instance, with sizes* d' *and* $\mathbf{d}$ *on bins of size* L *to integers gives an instance with sizes* L *and* $(\alpha+1)\mathbf{d}$ *on bins of size* $(\alpha+1)L$. *The new size now satisfies the assumption since* $(\alpha + 1) + 0 \leqslant L - 1$ *is implied by* $L \geqslant \alpha + 2$.

- *If a size already satisfies $\alpha_i + \beta_i \leqslant d_i - 1$, then after the transformation we have $\alpha_i' = \alpha_i$, $\beta_i' = (\alpha+1)\beta_i$ and $d_i' = (\alpha+1)d_i$, and thus $\alpha_i'+\beta_i' = \alpha_i+(\alpha+1)\beta_i \leqslant d_i-1+\alpha\beta_i \leqslant (\alpha+1)(d_i-1) \leqslant d_i'-1$. So that the assumption is still satisfied.*

Ergo, we can continue this process until there is no more i, such that $\alpha_i + \beta_i \geqslant d_i$, which proves the assumption. □

Multiplying the bound in assumption 3.16 by d_i gives the following corollary.

3.17 Corollary *If $\mathcal{L}^* \cong (m, L, \mathbf{d})$ then $\forall i\ L \leqslant (d_i - \beta_i)(d_i - 1)$, and in particular $L \leqslant d_m(d_m - 1)$*

Note that the transformation in effect rounds down the sizes that are close to unit fractions. Corollary 3.17 gives an interesting characterisation of max-gap instances as $\forall i\ d_i > \sqrt{L}$, which is useful if one wanted to enumerate instances for small values of L in search of a maximum gap instance.[4] Another observation one can make, is that in the max-gap examples there is no CTD-pattern in the optimal packing. This gives rise to the following lemma and implied transformation.

3.18 Lemma *If $\mathcal{L} \cong (m, L, \mathbf{d}, \mathbf{f})$ is a list which does not contain a CTD, then there exists a list $\mathcal{L}'$ with $\gamma(\mathcal{L}') \geqslant 1 - W_R(\mathcal{L}) + \frac{Mat(\mathcal{L})-1}{L-1}$, and this bound holds with equality when $W_R(\mathcal{L}) = 0$.*

Proof. *First scale the list so that all item sizes are integers and the bin size is L. The no-CTD requirement translates into the requirement that there is no $\mathbf{a} \leqslant \mathbf{f}$ such that $\langle \mathbf{a}, \mathbf{d} \rangle = L$. We will form the list $\mathcal{L}'$ by adding to $\mathcal{L}$ the minimal number r of items of size $x = 1 + \varepsilon$, such that $OPT(\mathcal{L}') = OPT(\mathcal{L}) + 1$.*

Suppose that $\mathcal{L}$ packs into an optimal configuration with wastage $w_j = L - \langle \mathbf{a}_j, \mathbf{d} \rangle$ in bin j. Note that $w_j \geqslant 1$ holds by the no-CTD assumption. Now choose x such that $\forall j\ \lfloor w_j/x \rfloor = w_j - 1$ and insert $\lfloor w_j/x \rfloor$ items x in bins $j = 1, \ldots, OPT(\mathcal{L})$. The total number of items inserted is given by $\sum \lfloor w_j/x \rfloor = \sum (w_j - 1) = OPT(\mathcal{L}) \times (L-1) - \langle \mathbf{f}, \mathbf{d} \rangle$, and is independent of the actual 'wastage/bin'-configuration. Ergo, r exceeds this number by one.

Choosing x such that $x \mid L$ gives $CSP_R(\mathcal{L}') \leqslant CSP_R(\mathcal{L}) + rx/L$ and the following lower bound.

$$\gamma(\mathcal{L}') = CSP_I(\mathcal{L}') - CSP_R(\mathcal{L}') \geqslant CSP_I(\mathcal{L}) + 1 - (CSP_R(\mathcal{L}) + rx/L) = \gamma(\mathcal{L}) + 1 - rx/L \qquad (3.8)$$

In order to maximise $\gamma(\mathcal{L}')$ we want to choose x minimal subject to $x \mid L$ and $x > 1$. This is achieved for $x = 1 + \frac{1}{L-1}$. This choice of x also satisfies $\forall j\ \lfloor w_j/x \rfloor = w_j - 1$. Substituting this

[4]The largest gap, found so far, can be derived from an instance (diagram 3.1) with (relatively) small bin-size $L = 60$.

(p,q)	L	$\mathbf{d}^T$	$\mathbf{f}^T$	γ
(2,3)	30	[15,10, 6]	[1,2,4]	$1\frac{1}{30} \approx 1.0333$
(3,4)	132	[44,33,12]	[2,3,6]	$1\frac{5}{132} \approx 1.0379$
(2,5)	90	[45,18,10]	[1,4,6]	$1\frac{1}{30} \approx 1.0333$
(3,5)	210	[70,42,15]	[2,4,7]	$1\frac{1}{30} \approx 1.0333$

Table 3.1. Listed are instances of harmonic CSPs, arising from (3.11), with $\gamma \geqslant 1\frac{1}{30}$. The first two examples correspond to the instances found by Fieldhouse, and Terno and Scheithauer.

in (3.8) together with the value for r gives

$$\begin{aligned}\gamma(\mathcal{L}') &\geqslant CSP_I(\mathcal{L}) - CSP_R(\mathcal{L}) + 1 - \tfrac{1}{L-1}\left[CSP_I(\mathcal{L}) \times (L-1) - \langle \mathbf{f}, \mathbf{d}\rangle + 1\right] = \\ & \quad 1 - \left[CSP_R(\mathcal{L}) - \langle \mathbf{f}, \mathbf{d}\rangle / L\right] + \tfrac{\langle \mathbf{f}, \mathbf{d}\rangle - L}{L(L-1)}\end{aligned} \tag{3.9}$$

and proves the first part of the lemma. For the second part note that if $CSP_R(\mathcal{L}) = Mat(\mathcal{L})$ then $CSP_R(\mathcal{L}') = Mat(\mathcal{L}) + rx/L$ and the bound for $\gamma(\mathcal{L}')$ holds with equality. □

Note that the transformation implied by the lemma creates a list $\mathcal{L}'$, such that

$$\mathcal{L}' \cong \Big(m+1,\ L(L-1),\ [(L-1)\mathbf{d}, L],\ [\mathbf{f}, OPT(\mathcal{L}) \times (L-1) - \langle \mathbf{f}, \mathbf{d}\rangle + 1]\Big) \tag{3.10}$$

The special case when all sizes in $\mathcal{L}$ are unit fractions gives rise to the following corollary.

3.19 Corollary *If $\mathcal{L}$ is harmonic and does not contain a CTD, then there exists a harmonic list $\mathcal{L}'$ with a gap $\gamma(\mathcal{L}') = 1 + \frac{Mat(\mathcal{L})-1}{L-1}$, where L is the (minimal) scalar for $\mathcal{L}$*

To illustrate this by an example we consider a list consisting of items of size $1/p$ and $1/q$, where $\gcd(p,q) = 1$, in frequencies f_p and f_q (this is the class of problems considered by Fieldhouse[25]). The list represents a harmonic CSP, so that $CSP_R = Mat = f_p/p + f_q/q$ and we can use corollary 3.19. The condition that there is no CTD implies that $f_p \leqslant p-1$ and $f_q \leqslant q-1$. In order to maximise $\gamma(\mathcal{L}')$ (in corollary 3.19) we need to maximise Mat, which is achieved by choosing $f_p = p-1$ and $f_q = q-1$. This gives $Mat = 2 - 1/p - 1/q$. It is easily seen that $OPT = 2$. The list $\mathcal{L}'$ now becomes as follows, with a gap of $\gamma(\mathcal{L}') = 1 + \frac{1-1/p-1/q}{pq-1}$ (see table 3.1 for specific examples).

$$\mathcal{L}' \cong (m', L', \mathbf{d}', \mathbf{f}') = \Big(3,\ pq(pq-1),\ [(pq-1)q, (pq-1)p, pq],\ [p-1, q-1, p+q-1]\Big) \tag{3.11}$$

All the examples with a gap larger than one, that have been published in literature (see section 3.1), have a small value for CSP_R (and CSP_I). This suggests that this is a characteristic of lists that do not possess the RU-property. This is not true. There is no lower bound for CSP_R such that all lists with a larger value for CSP_R have the RU-property, as shown by the following lemma.

3.20 Lemma $\forall c \, \exists \mathcal{L} \;\; CSP_R(\mathcal{L}) \geqslant c$ and $\gamma(\mathcal{L}) > 1$

Proof. *By construction. Choose* $n \in \mathbb{N}$ *such that* $n - \ln(n+1) \geqslant c$ *holds. Let* p_i *denote the* i^{th} *prime and construct a list with* $(p_i - 1)$ *items of size* $1/p_i$*, for* $1 \leqslant i \leqslant n$*. This list* $\mathcal{L}$ *does not contain a CTD, as is easily verified using congruences. Furthermore, it is harmonic so that* $CSP_R(\mathcal{L}) = Mat(\mathcal{L}) = \sum \frac{p_i - 1}{p_i} = n - \sum \frac{1}{p_i}$*. Since* $\sum \frac{1}{p_i} \leqslant \sum_{i=2}^{n+1} \frac{1}{i} < \ln(n+1)$ *we get a lower bound as* $CSP_R(\mathcal{L}) > n - \ln(n+1) \geqslant c$*. We now use corollary 3.19 to construct a list* $\mathcal{L}'$ *with* $CSP_R(\mathcal{L}') \geqslant CSP_R(\mathcal{L}) \geqslant c$ *and* $\gamma(\mathcal{L}') > 1$*. This proves the lemma.* □

Chapter 4

Harmonic CSP

4.1 Introduction

The *harmonic CSP* is defined as a cutting stock problem where all sizes divide the stock length an integral number of times. It can be scaled so that all the sizes are unit fractions. This means that we may assume that a harmonic CSP is defined on a list with $f_i \geqslant 1$ items of size $1/\alpha_i$, where $1 \leqslant i \leqslant m$ and $\alpha_i \in \mathbb{N}$, on a bin size of 1. As usual for a cutting stock problem we make the normalising assumption $0 < \alpha_1 < \cdots < \alpha_m$. We will also refer to a list as harmonic if all its items have sizes which are unit fractions.

It has appeared in the literature as the *self-deckling* CSP[19] and as a special case of the *self-dual* CSP.[25] Although its direct, practical applicability is limited, the harmonic CSP is important for the following reasons.

- Some of the largest known gaps, discovered up to date, lie within the class of harmonic cutting-stock problems. Studying this class might lead to insights for the more general problem.
- A good bound for [the gap of] the harmonic CSP provides us with an easily calculable bound for the normal CSP.

The LP-relaxation is easily solved; if f_i items of size $1/\alpha_i$ are required then the optimal LP-solution consists of patterns $\alpha_i \mathbf{e}_i$ used f_i/α_i times. So that $CSP_R = Mat = \sum f_i/\alpha_i$ holds for a harmonic cutting-stock problem.

We will study the duality gap of the harmonic CSP and are particularly interested in the structure of the lists that maximise it. First, formally define the harmonic gap and its parametric version as follows.

4.1 Definition $\gamma_{\rm h} \equiv \max\{OPT(\mathcal{L}) - \sum f_i/\alpha_i \mid \mathcal{L}$ *is harmonic*$\}$

4.2 Definition $\gamma_{\rm h}(\beta) \equiv \max\{OPT(\mathcal{L}) - \sum f_i/\alpha_i \mid \mathcal{L} \subset [\frac{1}{\beta}, 1]$ *is harmonic* $\}$

The analysis in this chapter is based on the following two properties. The first one is a simple consequence of corollary 3.14; by the normalising assumption we have that $\alpha_i \geqslant i$ and the property follows. The second one states that FFD is *near optimal* for a harmonic CSP. That is, it differs by at most one from the optimum solution value. This follows from lemma E5 (p. 242).

4.3 Property *The harmonic CSP has the NRU-property*

4.4 Property $\mathcal{L}$ *is harmonic* $\Rightarrow$ $FFD(\mathcal{L}) = \{0,1\} + OPT(\mathcal{L})$

The latter is a property that we will make extensive use of in the subsequent analysis. As a matter of fact it turns out that there are only a few instances, of the ones investigated in the subsequent analysis, for which the FFD-packing is not optimal.

4.2 Canonical form of maximum gap instances

Assume that $\mathcal{L}^*$ maximises $\gamma_{\rm h}(\mathcal{L})$ and that it consists of $f_i \geqslant 1$ items in the intervals $\langle\frac{1}{\alpha_i+1}, \frac{1}{\alpha_i}]$ with $0 < \alpha_1 < \cdots < \alpha_m$. For notational convenience denote $\alpha = \alpha_1$ and $\beta = \alpha_m$. This gives $\alpha_i \geqslant \alpha + i - 1$ and in particular $\beta \geqslant \alpha + m - 1 \geqslant m$. We may further assume that $\mathcal{L}^*$ packs into a singleton-bin configuration and that the smallest item in $\mathcal{L}^*$, $1/\beta$ is the singleton item (see chapter 3). This allows us to use bounds (3.2) and (3.3).

The structure of the harmonic CSP further allows us to make assumptions on the item frequencies f_i. In some cases it is possible to combine items and create a new, harmonic list with the same *Mat*-value. As an illustration consider the list $\{\frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{4}\}$, which FFD packs into 3 bins, but is easily seen to fit into 2 bins. We can combine the items $\frac{1}{4} + \frac{1}{4} \to \frac{1}{2}$ to create a new, harmonic CSP with the same *Mat*-value. This new list will pack in at least the same number of bins, so that it has a gap at least as large as the original list. To formalise this we define the following function.

4.5 Definition $\mathrm{sp}(i) \equiv$ *smallest prime factor of* i.

We can now make the following assumption on the item frequencies in $\mathcal{L}^*$.

4.6 Assumption $\forall i \;\; f_i \leqslant \mathrm{sp}(\alpha_i) - 1$

Proof. *Suppose that there is an i such that $f_i \geqslant \mathrm{sp}(\alpha_i)$. We can create a new CSP by removing $\mathrm{sp}(\alpha_i)$ items of size $1/\alpha_i$ and adding one item of size $\mathrm{sp}(\alpha_i)/\alpha_i$. This new CSP is also harmonic and has the same or larger gap.* □

A further assumption that can be made using the same rationale is that no subset of items sums to a unit fraction. As an example consider $\frac{1}{3} + \frac{1}{6} = \frac{1}{2}$, which implies that $\mathcal{L}^*$ does not have both an item of size $\frac{1}{3}$ and size $\frac{1}{6}$.

4.7 Assumption $\not\exists \mathbf{a} \leqslant \mathbf{f} \;\; \sum a_i/\alpha_i = 1/j,$ *in particular* $j = 1$

We shall use assumption 4.6 to derive upper bounds for γ_{h}. The last assumption is not easily incorporated in this derivation, but is useful to eliminate candidate lists, while searching for $\mathcal{L}^*$.

4.3 Upper bounds on harmonic gap

We start with another proof that the harmonic CSP has the NRU-property and refine this proof to eventually prove $\gamma_{\mathrm{h}} < 1\frac{1}{3}$.

By (3.2) we have the bound $OPT < 2 + \frac{Mat-1}{1-1/\beta}$, where $1/\beta$ is the smallest item in the list. Combining this with $Mat = \sum f_i/\alpha_i < m < \beta$ gives the following bound.

$$\gamma_{\mathrm{h}} < 2 + \frac{Mat - \beta}{\beta - 1} \;\Rightarrow\; \gamma_{\mathrm{h}} < 2 \tag{4.1}$$

We can sharpen the resulting bound for γ_{h} by using a better bound for Mat. To this end define $\mathcal{L}_\beta$ as the list with items of size $1/i$ in quantities $\mathrm{sp}(i) - 1$, for $2 \leqslant i \leqslant \beta$, and define N_β as the minimum number of bins the list $\mathcal{L}_\beta$ packs into. Now define the function $S(\beta)$ as follows.

4.8 Definition $S(\beta) \equiv \sum\limits_{i=2}^{\beta} \frac{\mathrm{sp}(i)-1}{i}$

This function and its properties are studied in appendix E.1. If $1/\beta$ is the smallest [critical] item size in $\mathcal{L}^*$ then $OPT(\mathcal{L}^*) \leqslant N_\beta$ and $Mat(\mathcal{L}^*) \leqslant S(\beta)$ hold. This is obvious since $\mathcal{L}^*$ is a subset of $\mathcal{L}_\beta$. We can now sharpen (4.1) using property E1, which bounds $S(\beta)$ by $\beta/2 - 1/\beta$, and derive the following bound.

$$\gamma_{\mathrm{h}} < 2 + \frac{S(\beta) - \beta}{\beta - 1} \;\Rightarrow\; \gamma_{\mathrm{h}} < 1\tfrac{1}{2} \tag{4.2}$$

The following lemma gives the last sharpening of the bound.

4.9 Lemma $\gamma_h < 1\frac{1}{3}$

Proof. *We split the range for β into two subranges.*

- *For $2 \leqslant \beta \leqslant 31$ we use the bound $\gamma_h < 1 + \frac{N_\beta - 2}{\beta}$, based on (3.3) and $OPT(\mathcal{L}^*) \leqslant N_\beta$. The values for N_β follow from tables 4.1, 4.2 and 4.3. Note that, with the exception of $\beta \in \{21, 23\}$, the packing by FFD is optimal for $\mathcal{L}_\beta$, see table 4.1. Optimal packings for these exceptions are given in tables 4.2 and 4.3, respectively. Using the values for N_β we can verify the following, where the last inequality is tight for $\beta = 24$.*

$$2 \leqslant \beta \leqslant 31: \qquad \gamma_h < 1 + \frac{N_\beta - 2}{\beta} \leqslant 1\tfrac{1}{3}$$

- *For $\beta \geqslant 32$ we can use property E2 to give*

$$\beta \geqslant 32: \qquad \gamma_h < 2 + \tfrac{S(\beta)-\beta}{\beta-1} \leqslant 2 + \tfrac{1+(\beta-1)/3-\beta}{\beta-1} \leqslant 1\tfrac{1}{3}$$

Combining the bounds for the subranges proves the lemma. □

Comment: One cannot expect to significantly reduce the bound in lemma 4.9, by further refining the above approach. To establish the bound $\gamma_h(\beta) < 1 + \frac{1}{k}$ one would need to prove the bound $S(\beta) \leqslant 1 + \frac{\beta-1}{k}$ for $\beta \geqslant \beta_0$, and verify that $\gamma_h(\beta) < 1 + \frac{1}{k}$ holds for $\beta < \beta_0$. The difficulty lies not so much in proving the first bound, but in the verification of cases for $\beta < \beta_0$.

As an illustration, taking $k = 4$ and consulting table E.1, one sees that β_0 is at least 100 and for $k = 5$ that β_0 is at least 400. Moreover, property E4 also shows that the upper bound for $\gamma_h(\beta)$, based on (4.1) is $1 + O(\frac{1}{\ln \beta})$. So that the number of cases one would have to check grows exponentially.

β	f_i	1	2	3	4	5	6	7	8	9	10	11	12	$S(\beta)$	*FFD*	opt	$1+\frac{N_\beta-2}{\beta}$
2	1	1												0.500	1	$\surd$	0.5
3	2	1	1											1.167	2	$\surd$	1
4	1	–	1											1.417	2	$\surd$	1
5	4	–	2	2										2.217	3	$\surd$	1.2
6	1	1	–	–										2.383	3	$\surd$	1.167
7	6	–	–	4	2									3.240	4	$\surd$	1.286
8	1	–	–	–	1									3.365	4	$\surd$	1.25
9	2	–	–	–	2									3.588	4	$\surd$	1.222
10	1	–	–	–	1									3.688	4	$\surd$	1.2
11	10	–	–	–	2	8								4.597	5	$\surd$	1.273
12	1	–	–	–	1	–								4.680	5	$\surd$	1.25
13	12	–	–	–	–	3	9							5.603	6	$\surd$	1.308
14	1	–	–	–	–	–	1							5.675	6	$\surd$	1.286
15	2	–	–	–	–	–	2							5.808	6	$\surd$	1.267
16	1	–	–	–	–	–	1							5.870	6	$\surd$	1.25
17	16	–	–	–	–	–	–	16						6.812	7	$\surd$	1.294
18	1	–	–	–	–	–	–	1						6.867	7	$\surd$	1.278
19	18	–	–	–	–	–	–	–	18					7.815	8	$\surd$	1.316
20	1	–	–	–	–	–	–	–	1					7.865	8	$\surd$	1.3
21	2	–	–	–	–	–	–	–	–	2				7.960	9		1.286
22	1	–	–	–	–	–	–	–	–	1				8.005	9	$\surd$	1.318
23	22	–	–	–	–	–	–	–	–	19	3			8.962	10		1.304
24	1	–	–	–	–	1	–	–	–	–	–			9.003	10	$\surd$	1.333
25	4	–	–	–	–	–	1	–	–	–	3			9.163	10	$\surd$	1.32
26	1	–	–	–	–	–	–	–	–	–	1			9.202	10	$\surd$	1.308
27	2	–	–	–	–	–	–	–	–	–	2			9.276	10	$\surd$	1.296
28	1	–	–	–	–	–	–	–	–	–	1			9.312	10	$\surd$	1.286
29	28	–	–	–	–	–	–	–	–	–	17	11		10.277	11	$\surd$	1.310
30	1	–	–	–	–	–	–	–	–	–	–	1		10.311	11	$\surd$	1.333
31	30	–	–	–	–	–	–	–	–	1	–	18	11	11.278	12	$\surd$	1.323

Table 4.1. The FFD-packing of $\mathcal{L}_\beta$ uses *FFD* bins. The optimal packing uses N_β bins.
Example: for $\beta = 4$ we have $\mathcal{L}_\beta = \{\frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{4}\}$. Discard rows 5–31, which eliminates all but columns 1 and 2 from the center of the table. These columns correspond to the two bins that FFD uses. The first bin contains items $\frac{1}{2}$ and $\frac{1}{3}$, while the second one contains items $\frac{1}{3}$ and $\frac{1}{4}$. Mutatis mutandis for other values of β. A tick indicates that the FFD-packing is optimal for that particular value of β.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	–	–	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2	–	1	1	1	–	–	–	1	–	–	–	–	–	–	–	–	1	–	1	–
3	–	–	–	3	–	2	–	–	–	–	–	–	–	1	–	–	–	–	–	1
4	–	–	–	–	–	2	1	1	–	–	1	–	–	1	–	2	–	4	–	–
5	–	–	–	–	–	1	–	–	–	3	–	4	–	–	–	3	–	1	–	1
6	–	–	–	–	–	–	–	–	1	5	–	2	–	–	1	3	–	1	–	–
7	–	–	–	–	–	1	–	–	–	–	–	1	–	–	–	7	–	7	–	–
8	–	–	–	–	–	–	–	–	–	2	–	5	1	–	–	1	–	5	–	–

Table 4.2. Optimal packing of $\mathcal{L}_{21}$ uses 8 bins. Column i represents the packing of the $f_i = \mathrm{sp}(i) - 1$ items of size $1/i$ over bins 1–8. Rows correspond to bins, e.g. bin 1 contains items 1/2, 1/3 and 1/6.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	1	–	–	1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2	–	1	1	1	–	–	–	1	–	–	–	–	–	–	–	–	1	–	1	–	–	–
3	–	–	–	3	–	2	–	–	–	–	–	–	–	1	–	–	–	–	–	1	–	–
4	–	–	–	–	–	3	–	–	–	5	–	–	1	–	–	–	–	–	–	–	1	–
5	–	–	–	–	–	–	–	1	–	–	–	7	–	–	1	4	–	1	–	–	–	–
6	–	–	–	–	–	1	–	–	1	–	1	1	–	1	–	1	–	4	–	–	–	6
7	–	–	–	–	–	–	1	–	–	1	–	–	–	–	–	8	–	1	–	–	–	6
8	–	–	–	–	–	–	–	–	–	2	–	1	–	–	–	3	–	9	–	1	–	1
9	–	–	–	–	–	–	–	–	–	2	–	3	–	–	–	–	–	3	–	–	–	9

Table 4.3. Optimal packing of $\mathcal{L}_{23}$ uses 9 bins. See legend of table 4.2 for details.

4.4 Bounds on parametric harmonic-gap

Having established an upper bound for γ_h, although crude, we now focus on finding instances of the harmonic CSP that have a gap larger than one. This is done by an enumerative approach. The results are listed in table 4.4 (more detailed results can be found in table E.4, p. 253).

β	1	2	3	4	5–10	11
$\gamma_h(\beta)$	0	1/2	2/3	3/4	31/30	137/132

Table 4.4. Parametric harmonic-gap for $\beta \leqslant 11$

Note that for $\beta \leqslant 4$ the gap is less than 1, so that all harmonic lists with smallest item $\geqslant 1/4$ have the RU-property. For $\beta \leqslant 10$ the largest gap $(1\frac{1}{30})$ is achieved by the list found by Fieldhouse.[24] For $\beta = 11$ there is only one list that achieves a larger gap and that is the list found by Scheithauer and Terno.[60] This list has a gap of $1\frac{5}{132}$.

Local bound for $\gamma_h(\beta)$

Since the number of cases one needs to verify grows exponentially with β, (table E.3) it is useful to have some sort of an estimate for the gap. To this end, suppose that we have a list $\mathcal{L}$ with smallest item $1/\beta$ and that it contains f_β of such items. Increase all these items from $1/\beta$ to $1/(\beta-1)$ to give the list $\mathcal{L}'$. We then have

$$OPT(\mathcal{L}) \leqslant OPT(\mathcal{L}') \leqslant \gamma_h(\beta-1) + Mat(\mathcal{L}') \leqslant \gamma_h(\beta-1) + \tfrac{f_\beta}{(\beta-1)\beta} + Mat(\mathcal{L}) \tag{4.3}$$

and the recursive bounds

$$\gamma_h(\beta) \leqslant \gamma_h(\beta-1) + \tfrac{\mathrm{sp}(\beta)-1}{(\beta-1)\beta} \quad \text{and} \quad \gamma_h(\beta) \leqslant \gamma_h(\alpha) + \sum_{i=\alpha+1}^{\beta} \tfrac{\mathrm{sp}(i)-1}{(i-1)i} \tag{4.4}$$

Combining this with the result for $\gamma_h(11)$ gives the following bounds for $\gamma_h(\beta)$.

β	12	13	14	15	16	17	18	19	20
$\gamma_h(\beta) \leqslant$	1.046	1.123	1.128	1.138	1.142	1.201	1.204	1.257	1.259

Table 4.5. Bounds on parametric harmonic-gap

Part II

Bin-packing Heuristics

Chapter 5

Bin-packing Heuristics

5.1 Introduction

The Bin-Packing Problem (BPP) can be formulated as follows;

> "given a list $\mathcal{L}$ of real numbers between 0 and 1, place the elements of $\mathcal{L}$ into a minimum number OPT of 'bins' so that no bin contains numbers whose sum exceeds 1."

It has applications in industry and business such as cutting stock and time tabling; in computer systems such as the allocation of files to segments on disks; in machine scheduling such as minimising the number of machines necessary for completing all tasks by a given deadline. These and other applications are discussed in Brown[7] and Johnson.[37]

Since the bin-packing problem is *NP-complete*, Garey and Johnson [31, p226], one can expect that in order to find optimal solutions a large amount of computation is necessary. For this reason, instead of trying to find optimal solution methods, one has concentrated on designing fast heuristic algorithms which provide a good approximate solution. In the case of the bin-packing problem several heuristics have been proposed and analysed with respect to their average- and worst-case behaviour.[4,38,52,55,72] Some of the better known heuristics are first-fit and first-fit decreasing. The worst-case performance of these are studied and discussed in Johnson.[37]

Generally, the methods used to establish a worst-case performance bound are based on a weighting function, suitably chosen for the particular heuristic under investigation (more on this in the next section). A discussion of some of the techniques and methods can be found in the literature.[11,12,27] Despite the emergence of these lines of attack; viz. problem reduction, area arguments and weighting functions, there is little common structure in the worst-case theory. This is illustrated by a quote from Fisher;[27]

> "After reviewing the results [on worst-case bounds] most readers will have obtained the impression that the worst-case theory of heuristics is quite fragmented, consisting of a collection of many problem specific results. This impression is absolutely correct. There is a clear need for a concept to put order to this chaos of specific results."

Coffman[11] starts his introductory remarks with

> "... the mathematics of bin-packing does not contain a central, well-structured theory that provides powerful, broadly applicable techniques for the analysis of approximation algorithms."

In particular on the topic of weighting functions he comments

> "It is common to hear the complaint that, although they are able to verify the correctness of the proof, the origins of the weighting functions, as well as the idea of the proof in the first place, remain shrouded in mystery."

and

> "Although a variety of guidelines can be formulated, at present a successful search for weighting functions normally must rely at some point on ingenuity (or luck)."

These comments were made well over a decade ago, but remain valid.

In this chapter we will outline a more structured approach to obtain bounds of the form $H \leqslant c + r\,CSP_R$, where c and r are some fixed constants. The key element in this approach is to identify the structure of the bins and patterns that can occur *more than once* in a realisation of the heuristic H. This concept of *recurrency* is used in subsequent chapters to derive worst-case bounds.

5.2 Notation & Terminology

In this section we have collected some more definitions and notations that are specific to bin-packing problems and heuristics.

By $\mathcal{L} = \{x_1, x_2, \ldots, x_n\}$ we denote a list of n items to be packed into bins of size 1. Each of the items in $\mathcal{L}$ has a *size* $x_i \in \langle 0, 1]$ associated with it. By $k\mathcal{L}$ we denote a list formed by concatenating

k copies of list $\mathcal{L}$. If two lists $\mathcal{L}$ and $\mathcal{L}'$ contain exactly the same items, so that after sorting [in say decreasing-size order] they are identical, we denote this by $\mathcal{L}' \cong \mathcal{L}$.

If the heuristic H packs the items into $N = H(\mathcal{L})$ bins, we can characterise the heuristic packing by a set of (binary) vectors $\mathbf{b}_1, \ldots, \mathbf{b}_N$, such that $\sum_{j=1}^{N} \mathbf{b}_j = \mathbf{e}$, and where b_{ij} denotes the indicator function for item i being packed into bin j. The set of all feasible bin-configurations $\mathcal{B}$ is given by $\mathcal{B} = \{\mathbf{b} \in \mathbb{B}^n | \langle \mathbf{b}, \mathbf{x} \rangle \leqslant 1\}$. Note that the elements of $\mathcal{B}$ are binary vectors.

An *on-line* heuristic is a heuristic which packs the items in the same sequence as they appear in the list. This models the situation where items 'arrive' one at a time and there is no prior knowledge of what the next item(s) will be. In the case of an *off-line* heuristic the entire list of items is known in advance. This allows a preprocessing [read reordering] of the list, after which it is packed by an on-line heuristic. Examples of on-line heuristics are next-fit and first-fit. Their off-line versions are next-fit decreasing and first-fit decreasing, where the list is sorted into decreasing sequence before being processed.

A *conservative* bin-packing heuristic opens a new bin if and only if the current item cannot be placed in the bins that are already active/open.

A bin-packing heuristic is *monotonic* if increasing the size of an item, or adding an item can never decrease the number of bins used by the heuristic (see also section E.5).

In a *fixed space* (or k-space) heuristic there can be at most k bins active at any given time. If an item cannot be placed and there are k bins active, one bin is closed and a new bin is opened in which the item is placed.

A heuristic [the bin configuration it packs a list into] is said to be *stable* if we can take any bin from the heuristic packing, delete all the items packed in this bin from the list, repack the resulting list and end up with exactly the same configuration (minus the bin we have deleted).

To avoid long-winded descriptions in the subsequent chapters we define the following concepts for notational brevity.

5.1 Definition ***(i-interval)*** *The interval* $\left\langle \frac{1}{i+1}, \frac{1}{i} \right]$

5.2 Definition ***(i-item)*** *An item with size in the interval* $\left\langle \frac{1}{i+1}, \frac{1}{i} \right]$

5.3 Definition ***(i-bin)*** *A bin where the largest item packed is an i-item.*

5.4 Definition ***(i-complete bin)*** *A bin packed with i (the maximal number of) i-items.*

5.3 Worst-case bounds

In the following chapters (6,7 and 8) we will study some well-known bin-packing heuristics. For each one of them a *performance* guarantee has been proven of the form

$$H(\mathcal{L}) \leqslant c + r\,OPT(\mathcal{L}), \tag{5.1}$$

where $H(\mathcal{L})$ is the number of bins the heuristic H uses, $OPT(\mathcal{L})$ is the minimal number of bins necessary to pack the list $\mathcal{L}$, and c and r are positive constants depending upon the particular heuristic used. We will say that a bound of the form (5.1) is *tight* if there are instances $\mathcal{L}$ for which it holds with equality. It is *asymptotically tight* if the constant r is the smallest possible.

The proof methods generally used to obtain bounds of this form are based upon a non-negative weighting function $W(x)$. Each item is assigned a weight $W(x_i)$ and the weight of a bin is thus calculated as $W(\mathbf{b}) = \sum_i W(x_i)\delta_i$, where δ_i is the indicator function for item i to be packed in this particular bin. The function $W(x)$ is chosen or rather, constructed[37] such that

1. "The total 'weight' of all the elements in the list $\mathcal{L}$ is no less than a fixed constant c short of the number of bins used in the particular packing under consideration."
2. "The total weight of any legally packed bin is less than some fixed constant r."

We can represent this formally as

$$W(\mathcal{L}) \geqslant H(\mathcal{L}) - c \tag{5.2}$$

$$W(\mathbf{b}) \leqslant r, \; \forall \; \mathbf{b} \in \mathcal{B} \tag{5.3}$$

Using the fact that for any [heuristic or optimal] packing of the list $\mathcal{L}$ the sum over the bin weights is equal to the sum over the item weights, i.e.

$$\sum_j W(\mathbf{b}_j) = W(\mathcal{L}) = \sum_i W(x_i) \tag{5.4}$$

allows us to combine (5.2) and (5.3) to establish (5.1);

$$H(\mathcal{L}) \leqslant c + W(\mathcal{L}) = c + \sum_j W(\mathbf{b}_j^*) \leqslant c + r\,OPT(\mathcal{L}), \tag{5.5}$$

where $\mathbf{b}_j^*$ represents the packing of bin j in the optimal packing and $1 \leqslant j \leqslant OPT(\mathcal{L})$.

Another, more natural way of deriving (5.1) is through the following identity, where W_j is the weight of bin j in the heuristic packing.

$$H(\mathcal{L}) = \sum_{j=1}^{H(\mathcal{L})} (1 - W_j) + W(\mathcal{L}) \tag{5.6}$$

Splitting the summation over bins with weight < 1 and bins with weight $\geqslant 1$ gives

$$H(\mathcal{L}) \leqslant \sum_{j|W_j<1} (1 - W_j) + W(\mathcal{L}) \tag{5.7}$$

5.4 Tightening of bounds

If a bound of the form (5.1) is derived by means of a weighting function as outlined in the previous section, we can immediately sharpen it. Instead of looking at the problem as a bin-packing problem, we can formulate it as a cutting stock problem, by grouping items of the same size. To bring it in the format of a CSP; suppose that L is a scalar for the list and that it contains m different item-types, with sizes $L \geqslant d_1 > \cdots > d_m > 0$, each occurring with frequency $f_1, \ldots, f_m$. We then have the following equivalence.

$$\mathsf{BPP}(\mathcal{L}) \cong \mathsf{CSP}(m, L, \mathbf{d}, \mathbf{f}) \tag{5.8}$$

Note that this does not imply any ordering of the list $\mathcal{L}$. A pattern in a solution to CSP can be characterised by a vector $\mathbf{a}$, indicating how many items of each item type can be 'cut' from the stock length. These vectors are elements of the *pattern set* $\mathcal{A}$, as defined in (2.2). The cutting stock problem corresponding to (5.8) and its LP-relaxation is given by (2.1) and (2.6) respectively. Solving the LP-relaxation gives a vector $\mathbf{z}$ which satisfies

$$\sum_j \mathbf{a}_j z_j \geqslant \mathbf{f} \quad \text{and} \quad CSP_R = \sum_j z_j \tag{5.9}$$

If (5.3) holds for all lists and all bin-configurations then any pattern in the pattern set $\mathcal{A}$ will also have a weight of at most r. So that

$$W(\mathbf{a}) \leqslant r, \ \forall\, \mathbf{a} \in \mathcal{A}. \tag{5.10}$$

Multiplying the first equation in (5.9) by the item weights w_i, and a subsequent summation yield

$$W(\mathcal{L}) = \sum_i f_i w_i \leqslant \sum_j W(\mathbf{a}_j) z_j \leqslant r \sum_j z_j = r\, CSP_R(\mathcal{L}). \tag{5.11}$$

We can now combine this with (5.2) to give the bound

$$H(\mathcal{L}) \leqslant c + r\,CSP_R(\mathcal{L}), \tag{5.12}$$

where $CSP_R(\mathcal{L})$ is the LP-relaxation of the bin-packing problem, formulated as a cutting stock problem. This is a sharper bound than (5.1) by virtue of the elementary bound.

The previous implies that any worst-case bound in terms of OPT, derived by means of a weighting function as described in section 5.3, can be replaced by a bound in terms of CSP_R.

Although for practical purposes this new bound does not make that much difference, since the difference (duality gap) between OPT and CSP_R is usually small (see chapter 3), it has theoretical advantages that will be used in the subsequent analysis. This stems from the fact that a linear program is *scalable*, viz. $CSP_R(k\mathcal{L}) = k\,CSP_R(\mathcal{L})$. We will exploit this feature in the next section.

5.5 Worst-case performance ratios

The standard measure for worst-case performance of a bin-packing heuristic is the *asymptotic worst-case performance ratio.* It is designed to give a characteristic of the 'steady state' behaviour of the heuristic when sufficient items have been packed, and the transient and start-up effects have become negligible. In this context one is interested in the worst-case behaviour of a heuristic in the 'long run'.

The *asymptotic* [worst-case performance] *ratio* R_H^∞ for a bin-packing heuristic H is usually defined,[17,37–39,41,43–46,71] with respect to the optimal integer solution as

$$R_H^\infty \equiv \limsup_{z\to\infty} \max_{\{\mathcal{L}|OPT(\mathcal{L})=z\}} \frac{H(\mathcal{L})}{z} \tag{5.13}$$

The *absolute* [worst-case performance] *ratio* R_H is defined as

$$R_H \equiv \sup_{\mathcal{L}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})} \tag{5.14}$$

To illustrate their meaning we take the next-fit (NF) heuristic (see appendix E.4) as an example. One can prove that for all lists $\mathcal{L}$ the bound $NF(\mathcal{L}) \leqslant -1 + 2OPT(\mathcal{L})$ holds. This proves $R_{NF}^\infty \leqslant 2$ and $R_{NF} \leqslant 2$. To prove that these bounds are tight, one can construct instances with arbitrary large values of $OPT(\mathcal{L})$ such that $NF(\mathcal{L}) = -1 + 2OPT(\mathcal{L})$. This proves $R_{NF}^\infty = R_{NF} = 2$.

If one considers only lists for which all the items have sizes not larger than φ; viz. $\mathcal{L} \subset \langle 0, \varphi]$, one can define the *parametric* [worst-case performance] *ratios*, $R_H^\infty(\varphi)$ and $R_H(\varphi)$ in an analogous manner.

5.5.1 A discussion of the asymptotic ratio

The asymptotic ratio gives a measure to rank the various heuristics; a heuristic with a smaller ratio [usually] gives a better performance. Although it is a good indicator to choose a certain heuristic for a particular application, on its own it does not give sufficient information. For practical purposes, what one is really after,[4,42,65] is a linear bound of the form

$$\forall \mathcal{L} \;\; H(\mathcal{L}) \leqslant c + r\,OPT(\mathcal{L}), \tag{5.15}$$

where c and r are constants. The asymptotic ratio is simply the minimal value of r such that this bound holds. If we take (5.15) as a starting point then this value of r is given by lemma E14, as

$$r_{\min} = \lim_{z \to \infty} \sup_{\{\mathcal{L} | OPT(\mathcal{L}) \geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})} \tag{5.16}$$

That $r_{\min}$ and R_H^∞ define the same number follows directly from the definition of lim sup.

Unit size

The asymptotic ratio is not independent of the assumption whether or not we are able to discriminate with infinite precision between the [sizes of the] items. Alternatively, it depends on whether or not we assume that all the item-sizes can be expressed in terms of [an integral multiple of] a unit size.

Consider the heuristic H, as a modification of the first-fit heuristic. It packs the items according to FF, with the added constraint that it places an i-item only in a bin which has other i-items, and if this is not possible, opens a new bin.[1] For a list $\mathcal{L}$ with $n_i \geqslant 1$ items, $i = 1, \ldots, m$ in the intervals $\langle \frac{1}{\alpha_i + 1}, \frac{1}{\alpha_i}]$, where $\alpha_i \in \mathbb{N}$ and $\alpha_1 < \cdots < \alpha_m$, the following is easily proved (the last bound follows from $\frac{1}{\alpha_i} < 2x_i$ for any item $x_i \in \langle \frac{1}{\alpha_i+1}, \frac{1}{\alpha_i}]$).

$$H(\mathcal{L}) = \sum_{i=1}^{m} \lceil \tfrac{n_i}{\alpha_i} \rceil, \;\; H(\mathcal{L}) \leqslant \sum_{i=1}^{m} \left(1 - \tfrac{1}{\alpha_i}\right) + \sum_{i=1}^{m} \tfrac{n_i}{\alpha_i} \;\; \text{and} \;\; H(\mathcal{L}) < m + 2Mat(\mathcal{L}).$$

[1]This example is not too far fetched, as heuristics that are based upon the principle of 'reserving' or dedicating [a fixed number of] bins for certain categories of items have appeared in the literature.[43]

Now consider the list $\mathcal{L}_m = \{1, \frac{1}{2}, \ldots, \frac{1}{m}\}$ which has the following characteristics; $H(\mathcal{L}_m) = m$, and $OPT(\mathcal{L}_m) \geqslant Mat(\mathcal{L}_m) = \sum_{i=1}^{m} 1/i > \ln m$. Applying the definition in (5.13) gives

$$R_H^\infty \equiv \limsup_{z\to\infty} \max_{\{\mathcal{L}|OPT(\mathcal{L})=z\}} \frac{H(\mathcal{L})}{z} \geqslant \limsup_{m\to\infty} \frac{H(\mathcal{L}_m)}{OPT(\mathcal{L}_m)} > \limsup_{m\to\infty} \frac{m}{\ln m} = \infty$$

However, if we are working in finite precision, that is the number of different item-types is finite (and thus $m \leqslant M$ for all lists, for some constant M) then

$$R_H^\infty \equiv \limsup_{z\to\infty} \max_{\{\mathcal{L}|OPT(\mathcal{L})=z\}} \frac{H(\mathcal{L})}{z} < \limsup_{z\to\infty} \frac{m + 2Mat(\mathcal{L})}{z} \leqslant \limsup_{z\to\infty} \frac{M}{z} + 2 = 2$$

So, depending on whether or not we assume to be able to discriminate with infinite precision between the [sizes of the] items, we can get two different answers to what the asymptotic ratio is.

Artificial instances

The instances that prove a lower bound for the asymptotic ratio can be very artificial. As an example consider the first-fit heuristic. The instances given by Johnson [37, pp. 302, 307] to show that the [parametric] asymptotic ratios can be approximated as closely as desired rely on items with sizes that are [in effect] a function of OPT and are smaller and smaller perturbations of unit fractions (see also comment 3 on page 61).

Practical relevance

There is a lack of relevance of the worst-case bounds to the problems that occur in practice. These bounds are exactly what they say they are: 'worst-case'. And if one happens to have a list which is not a worst-case instance this bound may be far too generous to be relevant. In practice one usually has a lot more information available than is put to use in the derivation of a performance bound.[2]

[2]Consider the example given in the introduction (p. 1) where one has to produce a number of items of length 3 and 4 from a stock length of 10. Suppose for purposes of illustration that the orders come in one at a time (without prior knowledge of the length of the order) and we have to make a decision there and then on how to allocate the order. A worst-case performance bound using FF on this example is $FF \leqslant \frac{5}{6} + \frac{7}{6}OPT$. This bound follows from (6.10) and the recurrent weighting function $W(4/10) = 1/2$ and $W(3/10) = 1/3$, which can be found solving the ratio program (5.24). That this bound is asymptotically tight follows from a list consisting of $6k$ items of size 4, followed by $12k$ items of size 3, which FF packs into $7k$ bins whereas the optimal packing uses $6k$ bins.

The general bound (6.2); $FF < 1 + 1.7OPT$ is far too generous to be of relevance in this case. Even the parametric bound in lemma 6.11; $FF \leqslant \frac{3}{2}OPT$, where we take into account that the largest item does not exceed $\frac{1}{2}$, does not remedy this.

Reference point

The performance bounds are expressed in OPT, but in some cases these bounds are merely a weaker version of the bounds that follow from their proofs. Consider for example the NF-heuristic where the bound $NF \leqslant 2OPT$, usually given in the literature,[11,12,28] originates from an argument based on the amount of material to be packed. This argument leads to the bound $NF \leqslant 2Mat$ from which the bound in OPT follows.

Although a bound in OPT tells us something about the range that the optimum solution can be in, for example $OPT \in [\frac{1}{2}NF, NF]$ for the NF-heuristic, it does not give any a priori information on the number of bins that we are actually using. This can be of use in certain applications.[3]

Generally though, it does not make sense to settle for a performance bound in terms of OPT, when there exists a tighter bound in terms of Mat (or CSP_R) from which the bound in OPT follows directly.

5.6 Recurrent ratio

The asymptotic ratio and bound are defined with respect to OPT, but as (5.12) suggests, it could be defined with respect to the optimal solution of the LP-relaxation of the associated CSP. However, rather than simply adapting the existing notation in section 5.5, we will derive an expression for the asymptotic ratio from first principle. As stated before, the ultimate objective of defining this ratio is to derive a linear bound. First we confine ourselves to one particular list.

Consider the following bound, where c and r are constants (depending upon $\mathcal{L}$), and where $k\mathcal{L}$ denotes a list constructed as k [concatenated] copies of list $\mathcal{L}$,

$$\forall k \in \mathbb{N}^+ \quad H(k\mathcal{L}) \leqslant c + r\, CSP_R(k\mathcal{L}). \tag{5.17}$$

That there exist constants c and r such that (5.17) holds can be seen as follows. The worst that a heuristic can do is to place every item in its own bin.[4] If there are n items in the list then clearly the bound $H(k\mathcal{L}) \leqslant kn$ holds, and one can choose $c = 0$ and $r = n/CSP_R(\mathcal{L})$.

The minimal value of r such that (5.17) holds is given by lemma E16, which leads us to define the *recurrent* asymptotic worst-case performance ratios (recurrent ratios for short) as follows.[5]

[3] Suppose that in cargo loading one quickly needs an estimate for the number of containers necessary. One cannot base the estimate upon a bound in OPT, since if one knew an optimal packing there would be no point in using a heuristic. A bound in Mat however can be used, since this quantity is readily calculated.

[4] We don't consider heuristics which are unnecessarily wasteful in that they open bins in which no items are placed.

[5] Note that $R_H^{rec}(\mathcal{L})$, defined with respect to OPT, gives the same value, since by corollary 3.6 we know that OPT

5.5 Definition $R_H^{rec}(\mathcal{L}) \equiv \limsup_{k\to\infty} \frac{H(k\mathcal{L})}{k\, CSP_R(\mathcal{L})}$

5.6 Definition $R_H^{rec} \equiv \sup_{\mathcal{L}} \; R_H^{rec}(\mathcal{L})$

This leads us to consider bounds of the following form, where $c(\mathcal{L})$ is a function with $c(k\mathcal{L}) = o(k)$ for all $\mathcal{L}$, that is $\forall\mathcal{L} \lim_{k\to\infty} c(k\mathcal{L})/k = 0$.

$$H(\mathcal{L}) \leqslant c(\mathcal{L}) + R_H^{rec} \times CSP_R(\mathcal{L}) \tag{5.18}$$

That bound (5.18) is the best possible, asymptotic bound in terms of CSP_R can be seen as follows. Suppose that there is a bound $\forall\mathcal{L}\;\; H(\mathcal{L}) \leqslant \bar{c}(\mathcal{L}) + \bar{r}\, CSP_R(\mathcal{L})$ with $\bar{c}(k\mathcal{L}) = o(k)$. Now take a list $k\mathcal{L}$ and let $k \to \infty$, which gives $R_H^{rec}(\mathcal{L}) \leqslant \bar{r}$ for all $\mathcal{L}$ and thus $R_H^{rec} \leqslant \bar{r}$.

If one considers only lists for which all the items have sizes not larger than φ; viz. $\mathcal{L} \subset \langle 0, \varphi]$, one can define the *parametric* recurrent ratio $R_H^{rec}(\varphi)$ in an analogous manner.

Recurrency

Having established an expression for the recurrent ratio, we now show that it is closely linked to the concept of recurrency. In the following, assume that we have one particular heuristic H to investigate. For any list $\mathcal{L}$ we can split the pattern-set $\mathcal{A}$ it generates, as defined by (2.2), into two disjunct sets; $\mathcal{A}_\infty$ and $\mathcal{A}_1$.

$$\mathcal{A}_\infty \equiv \{\mathbf{a} \in \mathcal{A} \mid \text{pattern } \mathbf{a} \text{ can occur more than once in a heuristic solution}\} \tag{5.19}$$

$$\mathcal{A}_1 \equiv \{\mathbf{a} \in \mathcal{A} \mid \text{pattern } \mathbf{a} \text{ can occur at most once in a heuristic solution}\} \tag{5.20}$$

We shall refer to these sets as the *recurrent* and *singular* (or non-recurrent) pattern set, respectively, and use A_∞ and A_1 to denote the matrix representation of these sets.

Assume that we have a list $\mathcal{L} \cong (m, L, \mathbf{d}, \mathbf{f})$, and denote by z^* the solution-value of $CSP_R(\mathcal{L})$. We are interested in lists $\mathcal{L}' \cong k\mathcal{L}$. Since any (finite) list can only generate a finite set of patterns it follows that, for k large enough, the packing of any such list $\mathcal{L}'$ must have patterns from $\mathcal{A}$ which occur a multiple number of times. Increasing k further will produce a heuristic solution where the fraction of patterns that occur only once will go to zero. This means that the recurrent ratio is determined by the recurrent pattern set. We now formalise this notion.

---

and CSP_R differ by at most m, the number of different item types in $\mathcal{L}$. Note further that it does not make sense to define (without further restrictions) an absolute ratio with respect to CSP_R. Take a list of one item of size $1/\alpha$, so that $H(\mathcal{L}) = 1$ and $CSP_R = 1/\alpha$ and thus $\sup H(\mathcal{L})/CSP_R(\mathcal{L}) = \infty$.

The heuristic packing of any list $\mathcal{L}' \cong k\mathcal{L}$ can be represented by two vectors $\mathbf{y}_1$ and $\mathbf{y}_\infty$, which represent the multiplicity of the corresponding patterns in the matrices A_1 and A_∞, respectively. The LP-relaxation of the cutting stock problem corresponding to $\mathcal{L}'$ can be represented by a vector $\mathbf{z}$, which represents the multiplicity of the corresponding patterns in the matrix A.

$$\mathsf{A}_1\mathbf{y}_1 + \mathsf{A}_\infty\mathbf{y}_\infty = k\mathbf{f} \quad \text{and} \quad H(\mathcal{L}') = \langle \mathbf{e}, \mathbf{y}_1 \rangle + \langle \mathbf{e}, \mathbf{y}_\infty \rangle \tag{5.21}$$

$$\mathsf{A}\mathbf{z} \geqslant k\mathbf{f} \quad \text{and} \quad \langle \mathbf{e}, \mathbf{z} \rangle = kz^* \tag{5.22}$$

We now determine an upper bound for the ratio $H(\mathcal{L}')/CSP_R(\mathcal{L}')$, by taking $\mathbf{y}_1$, $\mathbf{y}_\infty$ and $\mathbf{z}$ as decision variables. This gives the following MIP-formulation.

$$\frac{H(\mathcal{L}')}{CSP_R(\mathcal{L}')} \leqslant \boxed{\begin{array}{ll} \text{Max} & \langle \mathbf{e}, \mathbf{y}_1 \rangle / kz^* + \langle \mathbf{e}, \mathbf{y}_\infty \rangle / kz^* \\ \text{st} & \mathsf{A}_1\mathbf{y}_1 + \mathsf{A}_\infty\mathbf{y}_\infty = k\mathbf{f} \\ & \mathsf{A}\mathbf{z} \geqslant k\mathbf{f} \\ & \langle \mathbf{e}, \mathbf{z} \rangle = kz^* \\ & \mathbf{y}_1 \text{ binary, } \mathbf{y}_\infty \text{ integral} \\ & \mathbf{y}_\infty, \mathbf{z} \geqslant \mathbf{0} \end{array}} \leqslant |\mathcal{A}_1| / kz^* + \boxed{\begin{array}{ll} \text{Max} & \langle \mathbf{e}, \mathbf{y} \rangle \\ \text{st} & \mathsf{A}\mathbf{z} \geqslant \mathsf{A}_\infty \mathbf{y} \\ & \langle \mathbf{e}, \mathbf{z} \rangle = 1 \\ & \mathbf{y}, \mathbf{z} \geqslant \mathbf{0} \end{array}} \tag{5.23}$$

We now relax the first program as follows. The value of $\langle \mathbf{e}, \mathbf{y}_1 \rangle$ cannot exceed the cardinality of the non-recurrent pattern set, since by definition the components of $\mathbf{y}_1$ are either 0 or 1, and we can take the resulting upper bound out of the maximisation problem. Now combine the first two constraints to $\mathsf{A}\mathbf{z} \geqslant \mathsf{A}_\infty\mathbf{y}_\infty$, and the subsequent relaxing of the integrality of $\mathbf{y}_\infty$ allows us to scale the program. The dual of the resulting program is given by the following, *ratio program* [for a certain list].

$$\mathcal{R}(\mathcal{L}) = \boxed{\begin{array}{ll} \text{Min} & \rho \\ \text{st} & \langle \mathbf{a}, \mathbf{u} \rangle \leqslant \rho, \ \forall\, \mathbf{a} \in \mathcal{A} \\ & \langle \mathbf{a}, \mathbf{u} \rangle \geqslant 1, \ \forall\, \mathbf{a} \in \mathcal{A}_\infty \\ & \mathbf{u} \geqslant 0 \end{array}} \tag{5.24}$$

A bound for the recurrent ratio can now be obtained as follows.

$$R_H^{rec}(\mathcal{L}) \equiv \limsup_{k\to\infty} \frac{H(k\mathcal{L})}{CSP_R(k\mathcal{L})} \leqslant \limsup_{k\to\infty} |\mathcal{A}_1| / kz^* + \mathcal{R}(\mathcal{L}) = \mathcal{R}(\mathcal{L}) \tag{5.25}$$

and thus the bound

$$R_H^{rec} \leqslant \sup \mathcal{R}(\mathcal{L}). \tag{5.26}$$

Recap: We now have an explicit formulation in terms of a linear program, to determine [an upper bound for] the recurrent ratio of a heuristic. This LP is determined by the recurrent pattern set.[6] We note that the primal (5.23) [sometimes] allows us to construct a bin configuration that achieves this ratio, and thus prove that the upper bound for the recurrent ratio is tight.

Relationship to asymptotic ratio

We have tacitly worked on the assumption that the asymptotic ratio and recurrent ratio both define the same number. This is not the case as illustrated by the example of the heuristic on page 39, for which we have $R_H^{rec} < 2 \leqslant R_H^{\infty}$. However, it turns out that for many heuristics used in practice the two ratios are equivalent.

We now give some lemmas that can be used to prove the equivalence for a particular heuristic.

5.7 Lemma $R_H^{rec} \leqslant R_H^{\infty}$

Proof. *First note that $R_H^{rec}(\mathcal{L})$, defined with respect to CSP_R or OPT, give the same value since by corollary 3.6 we have that OPT and CSP_R differ by at most m, the number of different item types in $\mathcal{L}$. Now take one particular list $\mathcal{L}$.*

$$R_H^{rec}(\mathcal{L}) \equiv \limsup_{k\to\infty} \frac{H(k\mathcal{L})}{CSP_R(k\mathcal{L})} = \limsup_{k\to\infty} \frac{H(k\mathcal{L})}{OPT(k\mathcal{L})} = \lim_{k\to\infty} \sup_{t\geqslant k} \frac{H(t\mathcal{L})}{OPT(t\mathcal{L})}$$

We now extend the range over which the supremum is taken. For any list $t\mathcal{L}$ with $t \geqslant k$ we have $OPT(t\mathcal{L}) \geqslant CSP_R(t\mathcal{L}) \geqslant k CSP_R(\mathcal{L})$, so that $\{t\mathcal{L} \mid t \geqslant k\} \subset \{\mathcal{L}' \mid OPT(\mathcal{L}') \geqslant k CSP_R(\mathcal{L})\}$, and thus

$$R_H^{rec}(\mathcal{L}) \leqslant \lim_{k\to\infty} \sup_{\{\mathcal{L}' \mid OPT(\mathcal{L}') \geqslant k CSP_R(\mathcal{L})\}} \frac{H(\mathcal{L}')}{OPT(\mathcal{L}')} = \lim_{k\to\infty} \sup_{\{\mathcal{L}' \mid OPT(\mathcal{L}') \geqslant k\}} \frac{H(\mathcal{L}')}{OPT(\mathcal{L}')} = R_H^{\infty}$$

Since $CSP_R(\mathcal{L})$ is a constant it does not influence the 'lim sup' and we end up with (5.16), which is the equivalent definition of the asymptotic ratio, and proves the lemma. □

5.8 Lemma *If there is a constant c such that $\forall \mathcal{L}\ H(\mathcal{L}) \leqslant c + R_H^{rec} \times CSP_R(\mathcal{L})$ then $R_H^{rec} = R_H^{\infty}$*

Proof. *Follows directly by substitution in (5.13).*

$$R_H^{\infty} \leqslant \limsup_{z\to\infty} \max_{\{\mathcal{L} \mid OPT(\mathcal{L})=z\}} \frac{c + R_H^{rec} \times OPT(\mathcal{L})}{z} = R_H^{rec} + \limsup_{z\to\infty} c/z = R_H^{rec}$$

Combining this with lemma 5.7 proves the lemma. □

[6] If necessary one could extend the definition of recurrent patterns to patterns for which there is no upper bound on the number of times they can occur (hence the ∞-subscript in $\mathcal{A}_\infty$). This change does not affect the rationale and one would still end up with (5.24). However, the current definition of a recurrent pattern (5.19) suffices for the heuristics studied in this thesis.

5.9 Lemma *If there is a list $\mathcal{L}$ such that $H(k\mathcal{L}) = R_H^\infty \times kCSP_R(\mathcal{L})$ then $R_H^{rec} = R_H^\infty$*

Proof. *From definition 5.5 it follows directly that $R_H^{rec}(\mathcal{L}) = R_H^\infty$ and thus by definition 5.6 that $R_H^{rec} \geqslant R_H^\infty$. Combining this with lemma 5.7 proves the lemma.* □

5.10 Lemma *If there is a list $\mathcal{L}$ such that $H(k\mathcal{L}) = R_H^\infty \times kOPT(\mathcal{L})$ then $R_H^{rec} = R_H^\infty$*

Proof. *Same as of lemma 5.9, since the recurrent ratio can be defined with respect to OPT.* □

5.7 Weighting function

Ideally we would like to solve (5.24) for each list and then determine the maximum value over all lists. However this is not always practical or even feasible. Instead we determine a solution to (5.24) which is feasible for all lists. To this end we define a *recurrent weighting function* as follows.

5.11 Definition *A weighting function W is said to be recurrent [for a particular heuristic], if it is non-negative, non-decreasing and every recurrent pattern has weight of at least* 1.

Note that the first two requirements correspond to a property, listed in corollary 2.5, of the optimal dual multipliers of a CSP. In a sense a recurrent weighting function acts as a surrogate for the optimal dual multipliers. However, it provides us with a feasible solution to (5.24) for every list.

$$W \text{ is recurrent } \Rightarrow \mathcal{R}(\mathcal{L}) \leqslant \max\{W(\mathbf{a}) \mid \mathbf{a} \in \mathcal{A}\} \tag{5.27}$$

This relaxation is, in effect, the determination of the maximum pattern-weight over the pattern-set $\mathcal{A}$ for a particular list. The maximum of this over all lists gives an upper bound for the recurrent ratio as the following, *ratio program* [for a certain weighting function].

$$W \text{ is a recurrent weighting function for H} \Rightarrow R_H^{rec} \leqslant \sup \boxed{\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i \leqslant 1 \\ & x_i \geqslant 0 \end{array}} \tag{5.28}$$

5.8 Minimal list

We say that a list $\mathcal{L}$ *strongly dominates* a list $\mathcal{L}'$, if $\mathcal{L}'$ can be constructed from $\mathcal{L}$ by a sequence of one of the following operations;

1) delete an item,
2) reduce the size of an item,
3) cut an item into two or more items,

and denote this by $\mathcal{L}' \leqslant_{\mathsf{D}} \mathcal{L}$. Note that this leaves the order of the items intact (see also page 251). We call a list $\mathcal{L}$ *minimal*, with respect to a heuristic H, if

$$\forall \mathcal{L}' \leqslant_{\mathsf{D}} \mathcal{L} \quad H(\mathcal{L}') < H(\mathcal{L}) \tag{5.29}$$

We call an algorithm or function f, defined on a list, *strongly monotonic* if

$$\forall \mathcal{L}' \leqslant_{\mathsf{D}} \mathcal{L} \quad f(\mathcal{L}') \leqslant f(\mathcal{L}) \tag{5.30}$$

This implies that, when we consider the expression $H(\mathcal{L}) - rf(\mathcal{L})$, for a strongly monotonic function f, we only need to consider lists which are minimal with respect to H. Note that the functions Mat, CSP_{R} and OPT are all strongly monotonic.

5.9 Solution approach

In this section we will list and briefly discuss the main elements in the solution approach to derive a worst-case bound. It should be viewed as a collection of possible ingredients, rather than as a complete recipe.

Canonical form

Generally, these are assumptions that can be made on the sequence of items in the list. For instance; an algorithm may sort the items into decreasing order before processing, so that, without loss of generality (wlog), one may assume $x_1 \geqslant \cdots \geqslant x_n$. Another example are heuristics, such as first-fit, where one can reorder the items packed in a bin without affecting the packing [of the other bins], and thus make assumptions on the sequence of items in a bin.

Reference point

Instead of OPT we choose CSP_R (and sometimes Mat) as a reference point and consider the derivation of a worst-case bound as a maximisation problem, where $r > 1$ is a fixed constant.

$$c = \max_{\mathcal{L}} \; H(\mathcal{L}) - r\,CSP_R(\mathcal{L}), \tag{5.31}$$

Bin configuration

Determine what sort of bin configuration a heuristic produces and determine its characteristics.

- *Invariant:* A characteristic of the heuristic packing relating directly to the packing rule.
- *Recurrency:* A characteristic of a bin that occurs twice (or more) in a realisation of the heuristic.

Minimality

The fact that CSP_R is strongly monotonic implies that in determining the lists that maximise (5.31) we can restrict ourselves to minimal lists. Note that minimality implies the singleton-bin configuration (see diagram 3.2)

Weighting function

In the process of determining a weighting function there are two stages.

Ratio The structure of a recurrent bin leads to a recurrent weighting function W. We then determine the maximum pattern-value for this function. This gives a bound for the recurrent ratio. Obviously, we want this bound to be as tight as possible. We therefore want to choose a minimal [recurrent] weighting function, that is if we were to reduce the weight of any item size, the resulting weighting function would no longer be recurrent.

Constant Once we have determined the recurrent ratio r, we want to minimise the value of $c = \max_{\mathcal{L}} \; H(\mathcal{L}) - W(\mathcal{L})$ over all recurrent weighting functions that have r as the maximum pattern-weight. This implies that we are now looking for a maximal [recurrent] weighting function (subject to the condition that r is the maximum pattern-weight). This is usually done by perturbating/strengthening the weighting functions used in determining the ratio.

Instance

To prove that r is the recurrent ratio one constructs a family of lists, such that $H(k\mathcal{L}) = r\,CSP_R(k\mathcal{L})$ for all values of k. Having solved (5.28), the optimal patterns give an indication on how to construct examples that achieve the ratio r.

5.10 Worst-case bounds II

Although the values of c and r in a worst-case bound depend upon the particular heuristic used, there are some general statements that can be made. In the following, assume that we have a bound such that $\forall \mathcal{L}\ H(\mathcal{L}) < c + r\,CSP_R(\mathcal{L})$, and that c and r are constants.

5.12 Lemma *For a given $r \geqslant 1$, a bound of the form $H(\mathcal{L}) < 1 + r\,CSP_R(\mathcal{L})$ is the tightest bound [in terms of CSP_R] one can hope to derive for a bin-packing heuristic.*

Proof. *Take a list of one item of size $1/\beta$, so that $H = 1$ and $CSP_R = 1/\beta$, regardless of the heuristic chosen. This gives the requirement $\forall \beta\ c > 1 - r/\beta$, which implies that $c \geqslant 1$ must hold.* □

Note that in the lemma the requirement $r \geqslant 1$ is implicit in the assumption that a linear bound exists. Since $CSP_R \leqslant H < c + r\,CSP_R$ it follows that $\forall \mathcal{L}\ CSP_R < c + r\,CSP_R$ implies $r \geqslant 1$. We now tighten the lemma by excluding the trivial case $H(\mathcal{L}) = 1$ (or the equivalent $CSP_R(\mathcal{L}) \leqslant 1$ by the elementary bound).

5.13 Lemma *For a given $r \geqslant 1$ and lists with $H(\mathcal{L}) \geqslant 2$, the bound $H(\mathcal{L}) < 2 + r\,(CSP_R(\mathcal{L}) - 1)$ is the tightest bound [in terms of CSP_R] one can hope to derive for a bin-packing heuristic.*

Proof. *Take a list of $(\beta + 1)$ items of size $1/\beta$, so that $H \geqslant 2$ and $CSP_R = 1 + 1/\beta$, regardless of the heuristic chosen. This gives the requirement $\forall \beta\ c + r > 2 - r/\beta$, which implies that $c + r \geqslant 2$ must hold.* □

It is interesting to note that $c + r = 2$ can be proven to hold for some of the heuristics studied.

5.14 Lemma *If H is an on-line, conservative bin-packing heuristic then the instances that maximise $c(\mathcal{L}) = H(\mathcal{L}) - r\,OPT(\mathcal{L})$ satisfy $OPT(\mathcal{L}) \leqslant 1 - s + Mat(\mathcal{L})$, with s the smallest item in the list.*

Proof. *By corollary A2 we may assume that the list that maximises $c(\mathcal{L})$ consists of rationals, so that we can define $x = \gcd(1, x_1, \ldots, x_n)$.*

First assume that $W_I(\mathcal{L}) = OPT(\mathcal{L}) - Mat(\mathcal{L}) > 1$ and add $1/x$ items of size x to the front of the list to form the list $\mathcal{L}'$. Since H is on-line and conservative it follows that $H(\mathcal{L}') = 1 + H(\mathcal{L})$. We can use the optimal packing for $\mathcal{L}$ to produce a packing for $\mathcal{L}'$. If a bin has wastage w_i we can pack exactly w_i/x items of size x in this bin. Since the total wastage in the packing satisfies $W_I > 1$ it follows that we can pack all the additional items x and thus that $OPT(\mathcal{L}') = OPT(\mathcal{L})$. But this implies that $c(\mathcal{L}') = c(\mathcal{L}) + 1$ and therefore that $\mathcal{L}$ is not a maximising list. Ergo, the assumption $W_I(\mathcal{L}) > 1$ must be false.

Now assume that $1 - s < W_l(\mathcal{L}) \leqslant 1$ *and note that this implies* $W_l(\mathcal{L}) \geqslant 1 - s + x$. *In this case we add* $1 + (1 - s)/x$ *items of size* x *to the front of the list, and as in the first case we can prove that* $c(\mathcal{L}') = c(\mathcal{L}) + 1$ *and therefore that* $\mathcal{L}$ *is not a maximising list.*

So the assumption $W_l(\mathcal{L}) > 1 - s$ *must be false, which proves the lemma.* □

Note that lemma 5.14 implies that all lists that maximise $H(\mathcal{L}) - r\,OPT(\mathcal{L})$ have the RU-property.

5.15 Lemma *If H is an on-line, conservative bin-packing heuristic, and for all lists the bound* $H(\mathcal{L}) \leqslant c + r\,OPT(\mathcal{L})$ *holds, then the bound* $H(\mathcal{L}) \leqslant c - \left\lfloor \frac{OPT(\mathcal{L})+1}{L+2} \right\rfloor + r\,OPT(\mathcal{L})$ *also holds, where* L *is a scalar for the list* $\mathcal{L}$.

Proof. *The lemma is obvious for* $OPT(\mathcal{L}) \leqslant L$. *So, let* $\mathcal{L}$ *be a list such that* $OPT(\mathcal{L}) \geqslant L + 1$. *Now determine* $n = \left\lfloor \frac{OPT(\mathcal{L})-L-1}{L+2} \right\rfloor$ *and create a list* $\mathcal{L}'$ *by increasing the bin size from* 1 *to* $1 + \frac{1}{L+1}$, *and adding* $f = n \times (L^2 + 2L) + (L^2 + L)$ *items of size* $\varepsilon = \frac{1}{L(L+1)}$ *to the front of the list* $\mathcal{L}$.

- *Every bin in the optimal packing of* $\mathcal{L}$ *can accommodate* L *items* ε, *after we have increased the bin size to* $1 + L\varepsilon$. *It is easily verified that* $f \leqslant OPT \times L$, *so that all items* ε *can be placed, and thus* $OPT(\mathcal{L}') = OPT(\mathcal{L})$.
- *The* f *items* ε *will pack in* $n + 1$ *bins. The first* n *bins take* $(L^2 + 2L)$ *items each and are thus filled completely. Bin* $(n + 1)$ *is filled to a level of exactly* 1 *and thus has wastage* $\frac{1}{L+1}$. *Since all items in* $\mathcal{L}$ *are* $\geqslant \frac{1}{L}$, *it follows that none of these items is placed in any of the bins* $1, \ldots, n + 1$ *and thus that bins* $n + 2, \ldots$ *are packed exactly as* $\mathcal{L}$ *is packed. Ergo* $H(\mathcal{L}') = n + 1 + H(\mathcal{L})$.

Since the bound $H(\mathcal{L}') \leqslant c + r\,OPT(\mathcal{L}')$ *holds by assumption, it follows that the bound* $H(\mathcal{L}) \leqslant c - (n + 1) + r\,OPT(\mathcal{L})$ *also holds. Substituting the value for* n *now proves the lemma.* □

5.16 Corollary *If H is an on-line, conservative bin-packing heuristic, then any instance that maximises* $H(\mathcal{L}) - r\,OPT(\mathcal{L})$ *satisfies* $OPT(\mathcal{L}) \leqslant L$, *where* L *is a scalar for the list* $\mathcal{L}$.

Chapter 6

First-Fit Heuristic

6.1 Introduction

The *First-Fit (FF)* heuristic takes a list $\mathcal{L}$ and places each item, in succession, into the first bin in which it fits. When an item cannot be placed, a new bin is opened in which this item is placed. As an illustration we give the following example [1]

$$\text{List } \mathcal{L} = \{\ \underbrace{1,\ldots,1}_{270},\ \underbrace{45,\ldots,45}_{12},\ \underbrace{42,\ldots,42}_{5},\ \underbrace{104,\ldots,104}_{20},\ \underbrace{156,\ldots,156}_{20}\ \} \text{ on bins of size 311.}$$

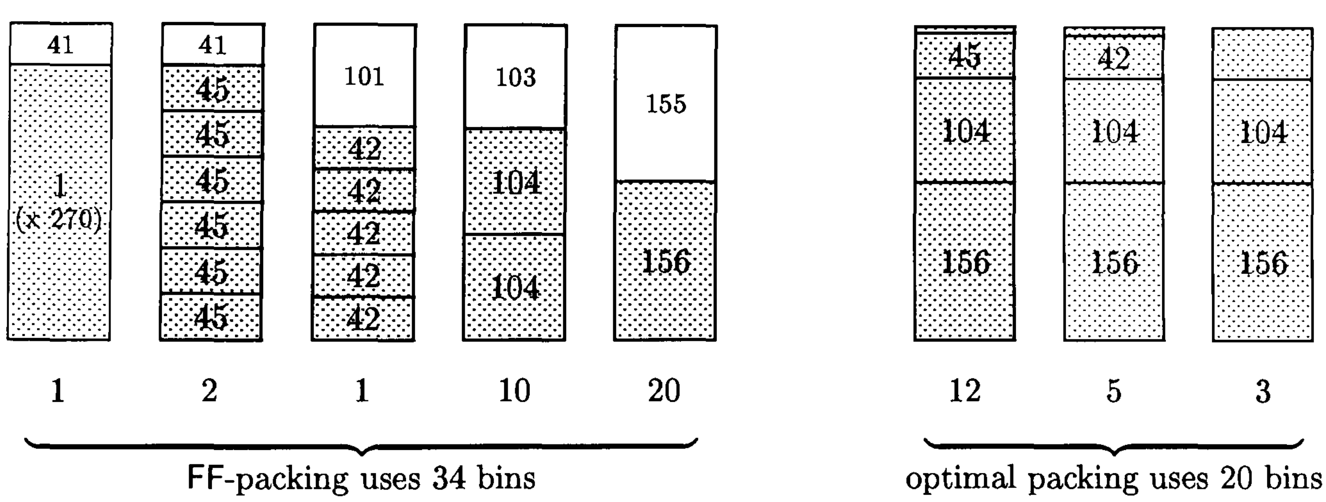


Diagram 6.1. An example of a packing by first-fit.

The FF-heuristic was studied extensively by Johnson.[38] Its worst-case performance was determined using a weighting function as outlined in section 5.3. The main result given in a summary article

[1] An example with $FF = 17$ and $OPT = 10$ can be found in Johnson.[37]

by Johnson et al.[37] regarding first-fit is

$$FF \leqslant 2 + 1.7OPT \tag{6.1}$$

It was further shown, by means of a construction, that there are lists which will give $FF = 17k$ and $OPT = 10k + 1$, for every $k \in \mathbb{N}^+$. This implies that $R^\infty_{FF} = 1.7$.

In a subsequent article Garey et al.[30] noted that one could improve slightly upon the constant to give

$$FF < 1 + 1.7OPT \tag{6.2}$$

It is worth mentioning that this appeared almost as a sideline comment in the article and possibly the cause that (6.2) is not common knowledge. In some later publications[49, 52] bound (6.1) is quoted instead of (6.2).

In this chapter we will present a proof, based on the concepts developed in chapter 5, to sharpen the FF-bound to

$$FF < 1 + 1.7CSP_R \tag{6.3}$$

where CSP_R is the value of the LP-relaxation of the bin-packing problem, formulated as a cutting stock problem. For more restricted lists we will improve the upper bound $FF \leqslant 2 + \frac{\alpha+1}{\alpha}OPT$, given by Johnson,[37] to

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha} \right] \;\Rightarrow\; FF < \tfrac{\alpha-1}{\alpha} + \tfrac{\alpha+1}{\alpha}Mat \tag{6.4}$$

where φ is the largest item in the list, $\alpha = \lfloor 1/\varphi \rfloor$ and $Mat = \sum x_i$. Furthermore, we will show that this bound is the best possible.

6.2 Canonical form

Suppose that we have packed a list $\mathcal{L} = \{x_1, \ldots, x_n\}$ using the FF-heuristic. Wlog we may assume that the list consists of consecutive blocks of items corresponding to the bins into which they are packed, and that each block consists of a sequence of items of non-increasing size. Additionally, for a minimal list we may assume that the last bin is a singleton bin; deleting all but one of the items in this bin will give a list which packs into the same number of bins. The FF-rule leads directly to an 'invariant', which any packing produced by first-fit will satisfy:

$$\textbf{FF-invariant:}\quad \forall i < j \;\; w_i < s_j, \tag{6.5}$$

where w_i is the wastage in bin i and s_j is the smallest item in bin j.

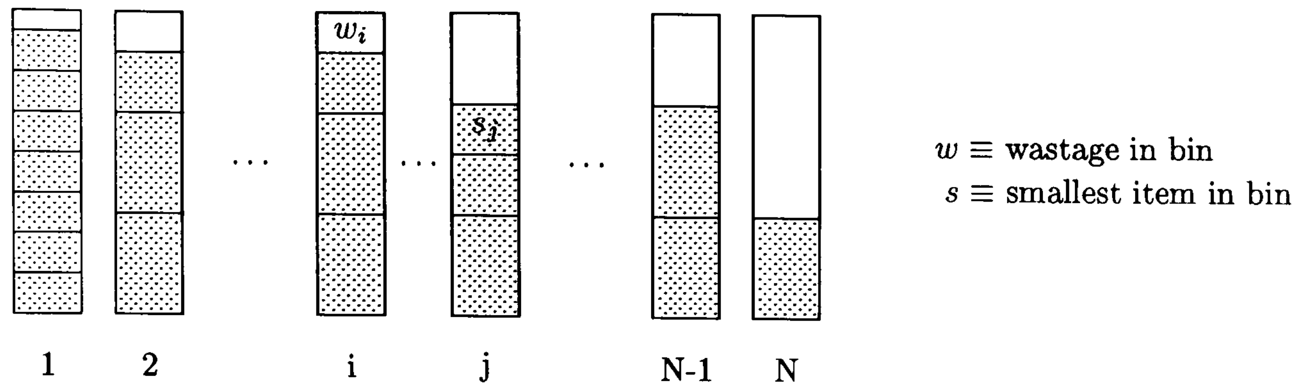


Diagram 6.2. Minimal FF-configuration.

6.3 Recurrent patterns

To use the analysis in chapter 5 we need to determine the structure of patterns that can occur more than once in a realisation of FF. The next lemma gives a necessary condition for such a pattern.

6.1 Lemma *Any bin that occurs more than once in a packing by first-fit has $s > w$.*

Proof. *By contradiction. Assume there is a bin which has $s \leqslant w$ and occurs twice. Label these bins as 1 and 2 (so that 1 precedes 2). Under the assumption $s \leqslant w$, the smallest item in bin 2 will fit in bin 1. But this means that FF could have packed this item in bin 1, and therefore that FF would have never placed it in bin 2. This leads to a contradiction and proves the lemma.* □

6.2 Lemma *Any set of N bins, each bin with $s_j > w_j$, can be combined to a list which first-fit packs into a configuration using N bins.*

Proof. *Sort the bins in order of increasing wastage, so that $i \leqslant j$ implies $w_i \leqslant w_j$. The resulting bin-configuration constitutes a valid packing by first-fit, since $s_j > w_j \geqslant w_i$, for all $i \leqslant j$. The corresponding list consists of consecutive blocks of items, corresponding to the sorted sequence of bins. Obviously, this list packs into N bins.* □

The previous two lemmas lead to the following definition.

6.3 Definition ***(FF-recurrent bin)*** *A bin containing m items, with sizes $1 \geqslant x_1 \geqslant \cdots \geqslant x_m > 0$ such that $\sum_{i=1}^{m} x_i > 1 - x_m$, is said to be FF-recurrent.*

6.4 Lemma *A FF-recurrent bin with largest item in the interval $\langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ contains at least α items.*

Proof. *Assume that the bin contains m items. Let x_1 be the largest and x_m be the smallest item. By definition of recurrency we have $\sum x_i > 1 - x_m$. From $m x_1 \geqslant \sum x_i > 1 - x_m \geqslant 1 - x_1$ it follows that $m > 1/x_1 - 1 \geqslant \alpha - 1$ and thus that $m \geqslant \alpha$. This proves the lemma.* □

6.4 General bound

Suppose that we have a recurrent weighting function W. We now derive a lower bound for the bin weight of non-recurrent bins. Let ε be such that $1/\varepsilon$ is a scalar for all the item sizes x_i. Such an ε

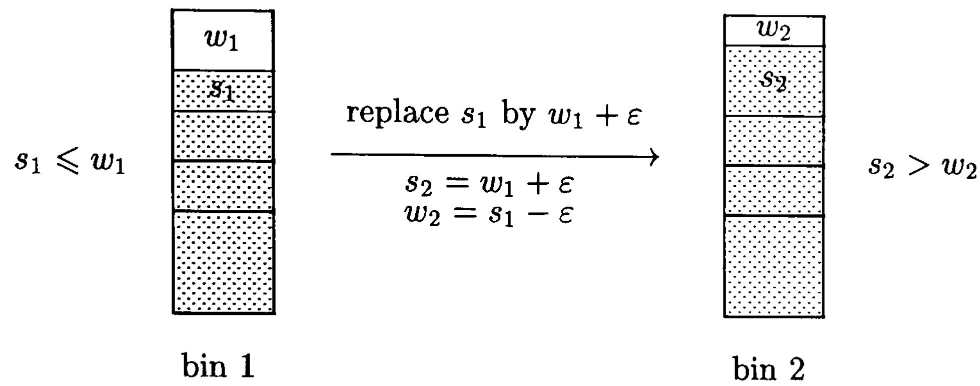


Diagram 6.3. Lower bound for the bin weight of non-recurrent bins under FF.

exists, see appendix A. This implies that the inequality $s > w$ can be replaced by $s \geqslant w + \varepsilon$. Now take bin 1, which has $s_1 \leqslant w_1$, as in diagram 6.3 and create a new bin configuration by replacing an item with size s_1 by an item with size $w_1 + \varepsilon$. The new bin has a wastage $w_2 = s_1 - \varepsilon$. The smallest item in this bin, s_2 satisfies $s_2 \geqslant \min(s_1, w_1 + \varepsilon) = s_1$. So, for the new bin $s_2 > w_2$ holds, which means that for the weight of bin 2 we have $W_2 \geqslant 1$, since the function W is recurrent. Using the identity $W_2 = W_1 - W(s_1) + W(w_1 + \varepsilon)$ yields the bound

$$s \leqslant w \;\Rightarrow\; W_{\text{bin}} \geqslant 1 - [W(w + \varepsilon) - W(s)]. \tag{6.6}$$

Now consider $c(\mathcal{L}) = FF(\mathcal{L}) - W(\mathcal{L})$, and denote by W_j the weight of bin j, then

$$c(\mathcal{L}) \;=\; \sum_j (1 - W_j) \;\leqslant\; \sum_{j|W_j<1} (1 - W_j) \tag{6.7}$$

Wlog we may assume that the list that maximises $c(\mathcal{L})$ is packed by FF into N bins, all with bin weight strictly less than one. Since FF is stable, we can delete [the items in] the bins with bin weight $\geqslant 1$ to create a new list with a larger or equal c-value. If a bin has weight $W_j < 1$ then $s_j \leqslant w_j$ must hold and we can apply (6.6) to lower bound its weight.

$$c(\mathcal{L}) \;\leqslant\; \sum_{j=1}^{N-1} (1 - W_j) + 1 - W_N \;\leqslant\; \sum_{j=1}^{N-1} [W(w_j + \varepsilon) - W(s_j)] + 1 - W(s_N) \tag{6.8}$$

Using the FF-invariant which implies $w_j + \varepsilon \leqslant s_{j+1}$, and the fact that W is non-decreasing gives

$$c(\mathcal{L}) \;\leqslant\; \sum_{j=1}^{N-1} [W(s_{j+1}) - W(s_j)] + 1 - W(s_N) \;=\; 1 - W(s_1). \tag{6.9}$$

This implies that $FF \leqslant 1 - W(s_1) + W(\mathcal{L})$, where s_1 is the smallest item in the first bin with bin weight strictly less than 1. If there is no bin with bin weight < 1 then $FF \leqslant W(\mathcal{L})$ holds by (6.7). Since $W(s_1) > 0$ we arrive at the following bound.

$$W \text{ is recurrent } \Rightarrow FF < 1 + W(\mathcal{L}) \tag{6.10}$$

6.5 Weighting function

In order to choose a suitable (recurrent) weighting function W we will derive further requirements by considering the following bin configurations.

1) Bin filled with $\alpha = \lfloor 1/x \rfloor$ items of size x.

   The corresponding pattern has $s > w$ and is therefore recurrent. This leads to the following requirement.

$$\alpha W(x) \geqslant 1, \text{ for } x \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha} \right] \tag{6.11}$$

2) Bin filled with one item of size x and one item of size $\left(\frac{1-x}{2}\right)^+$, where $x \in \left\langle \frac{1}{3}, \frac{1}{2} \right]$.

   The corresponding pattern is also recurrent and this translates into the following requirement.

$$W(x) + W(\tfrac{1-x}{2}) \geqslant 1, \text{ for } x \in \left\langle \tfrac{1}{3}, \tfrac{1}{2} \right] \tag{6.12}$$

These two requirements are sufficient to construct a recurrent weighting function. Starting with (6.11) we rewrite it as $W\left(\frac{1+\Delta}{\alpha+1}\right) \geqslant \frac{1}{\alpha}$, for $\Delta \in \left\langle 0, \frac{1}{\alpha} \right]$. Now substitute $x = \frac{1+\Delta}{\alpha+1}$ to yield $W(x) \geqslant \frac{x}{1+\Delta-x}$. This naturally leads to $\frac{x}{1-x}$ as a weighting function. This function is recurrent since for $x_i \geqslant x_m$ and $\sum x_i > 1 - x_m$ we have

$$\sum_i W(x_i) = \sum_i \frac{x_i}{1-x_i} \geqslant \min_i \left(\frac{1}{1-x_i}\right) \sum_i x_i \geqslant \frac{1}{1-x_m} \sum_i x_i > 1 \tag{6.13}$$

Take $\frac{x}{1-x}$ as a weighting function for $x \leqslant \frac{1}{3}$, and (6.12) directly gives the requirement $W(x) \geqslant \frac{2x}{1+x}$ for $x \in \left\langle \frac{1}{3}, \frac{1}{2} \right]$. Combining the requirements gives the following weighting function.

$$W(x) = \begin{cases} \frac{x}{1-x}, & 0 \leqslant x \leqslant \frac{1}{3} \\ \frac{2x}{1+x}, & \frac{1}{3} < x \leqslant \frac{1}{2} \\ 1, & \frac{1}{2} < x \leqslant 1 \end{cases} \tag{6.14}$$

This function and the gridpoints $\left(\frac{1}{\alpha+1}, \frac{1}{\alpha}\right)$ are depicted in diagram 6.4. We now prove that this weighting function is recurrent.

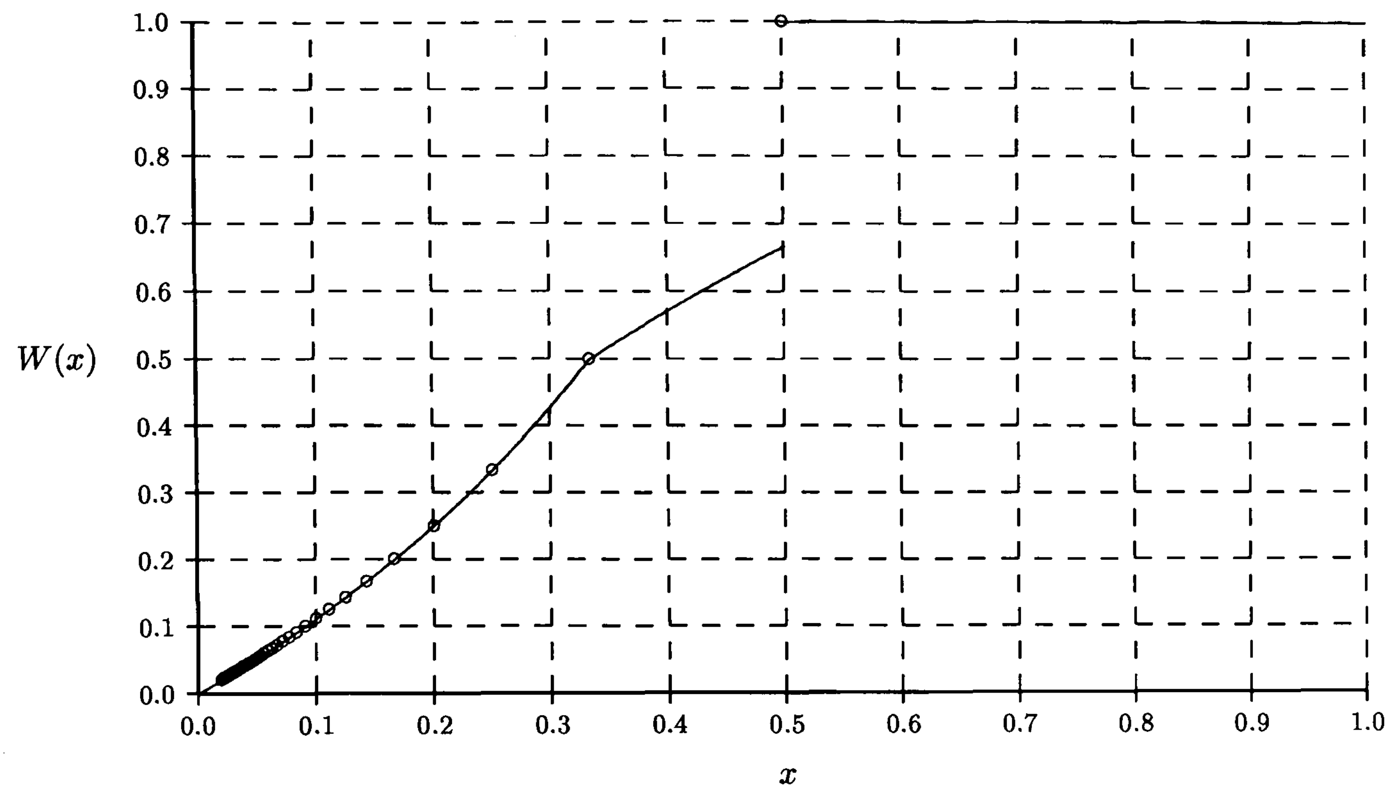


Diagram 6.4. The recurrent weighting function (6.14) for first-fit.

6.5 Claim *$W(x)$ as defined in (6.14) is recurrent*

Proof. *Assume that we have a recurrent bin with items $x_1 \geqslant \cdots \geqslant x_m$, that this bin has the smallest possible weight, and that this weight is strictly less than one. We can make the following assumptions on the items in the bin.*

(i) There is no 1-item.

(ii) There is exactly one 2-item.

If there are two then $W_{\text{bin}} \geqslant 1$. If there is none, then $W_{\text{bin}} \geqslant 1$ by the recurrency of $\frac{x}{1-x}$.

(iii) All the items $\leqslant 1/3$ have the same size.

Otherwise, by the strict convexity of W on $\langle 0, 1/3]$, we can create a bin with smaller weight.

Now denote by z the size of the 2-item, by y the size of the items $\leqslant 1/3$ and by n the number of items y. We can now formulate the weight of the bin as the following program.

$$W_{\text{bin}} = \boxed{\begin{array}{ll} \text{Min} & W(z) + nW(y) \\ \text{st} & z + ny > 1 - y \\ & y \in \langle 0, \tfrac{1}{3}] \text{ and } z \in \langle \tfrac{1}{3}, \tfrac{1}{2}] \\ & n \in \mathbb{N}^+ \end{array}} \qquad (6.15)$$

Note that $n \in \mathbb{N}^+$, since $n = 0$ leads to an infeasibility. The constraint directly gives a lower bound for y, so that we can lower bound the term $nW(y)$ in the objective function.

$$nW(y) > nW\left(\tfrac{1-z}{n+1}\right) = n\tfrac{1-z}{n+z} \geqslant \tfrac{1-z}{1+z} \quad \Rightarrow \quad W(z) + nW(y) > \tfrac{2z}{1+z} + \tfrac{1-z}{1+z} = 1 \qquad (6.16)$$

and thus $W_{\text{bin}} > 1$. This leads to a contradiction. Therefore $W_{\text{bin}} \geqslant 1$, which proves the claim. □

6.6 Recurrent ratio

6.6 Claim *The maximum pattern-value under $W(x)$, as defined in (6.14), is strictly less than 1.7*

Proof. *This value can be expressed as $r = \max \left\{\sum W(x_i) \mid \sum x_i \leqslant 1,\ x_i \geqslant 0\right\}$.*

The configuration $\left\{\frac{1}{2}+\varepsilon, \frac{1}{3}, \frac{1}{6}-\varepsilon\right\}$ gives a lower bound as $r \geqslant 1.7 - \frac{36\varepsilon}{25+30\varepsilon}$, for $\varepsilon > 0$ sufficiently small. This implies that for an optimal solution we may assume

i) There is a 1-item active in the optimal solution and this item has size $\frac{1}{2}+\varepsilon$. If there is no 1-item active, then $r \leqslant \max_{x \leqslant 1/2}\left(\frac{W(x)}{x}\right) \times 1 = 3/2$. Since $W(x) = 1$ for $x > 1/2$ we may assume that the 1-item active is the smallest item larger than $1/2$.

ii) If there are two items $\leqslant 1/3$ active then the largest one has size $1/3$. This follows from the convexity of $W(x)$ on $\langle 0, 1/3]$

We now prove the claim by a case analysis on the 2-items in the solution.

1) If there is a 2-item active of size y, then there is also one item active of size $\frac{1}{2}-\varepsilon-y$. This follows from the convexity of W on $\langle 0, 1/3]$. The maximum bin weight in this case is given by $1 + W(y) + W(\frac{1}{2}-\varepsilon-y)$ with $1/3 < y \leqslant 1/2$. This is strictly upper bounded by the function

$$f(y) = 1 + W(y) + W(\tfrac{1}{2}-y) = 1 + \frac{2y}{1+y} + \frac{\frac{1}{2}-y}{\frac{1}{2}+y}. \qquad (6.17)$$

The function $f(y)$ is maximised on the interval $[1/3, 1/2]$ for $y = 1/3$ with a value of $f(1/3) = 1.7$. So, for this case we have $r < 1.7$.

2) If there is no 2-item active, then the convexity of W on $\langle 0, \frac{1}{3}]$ implies that the maximum is attained for an item of size $1/3$ and one of size $\frac{1}{6}-\varepsilon$ together with the item of size $\frac{1}{2}+\varepsilon$. This is the previously mentioned configuration with value strictly less than 1.7.

Ergo, the maximum pattern-value under weighting function (6.14) is strictly less than 1.7. □

We can now combine the claim with (6.10) to establish (6.3). It further proves that $\forall \mathcal{L}\ R^{rec}_{FF}(\mathcal{L}) < 1.7$ and thus that $R^{rec}_{FF} \leqslant 1.7$.

6.7 Lists with no 1-items

Let φ be the largest item in the list. For $\varphi \leqslant 1/2$ the following bounds are given by Johnson.[37]

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha}\right] \;\Rightarrow\; FF \leqslant 2 + \tfrac{\alpha+1}{\alpha} OPT \tag{6.18}$$

The constant can be improved upon using (6.10) and the weighting function $W(x) = \frac{\alpha+1}{\alpha}x$ to give the following bound, where $Mat = \sum x_i$.

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha}\right] \;\Rightarrow\; FF < 1 + \tfrac{\alpha+1}{\alpha} Mat \tag{6.19}$$

To prove (6.19) it suffices to show that the weighting function is recurrent.

6.7 Lemma *$W(x) = \frac{\alpha+1}{\alpha}x$ is recurrent for $\mathcal{L} \subset \langle 0, \frac{1}{\alpha}]$.*

Proof. *Let x_m be the smallest item in a recurrent bin, so that $\sum x_i > 1 - x_m$ holds. Now distinguish between the following two cases with regard to x_m.*

$x_m \leqslant \frac{1}{\alpha+1}$: $W_{\text{bin}} = \sum W(x_i) = \frac{\alpha+1}{\alpha}\sum x_i > \frac{\alpha+1}{\alpha}(1 - x_m) \geqslant 1$

$x_m > \frac{1}{\alpha+1}$: *$m \geqslant \alpha$ by lemma 6.4, so that* $W_{\text{bin}} = \sum W(x_i) \geqslant m \times W(x_m) > \alpha \times W\left(\frac{1}{\alpha+1}\right) = 1$

So for both cases the bin weight of a recurrent bin is at least 1, which proves the lemma. □

We can sharpen (6.19) further by a more detailed analysis on the bin configuration, as is done by the following lemmas.

6.8 Lemma $\varphi \in \left\langle \frac{1}{\alpha+1}, \frac{1}{\alpha}\right] \Rightarrow FF < \frac{\alpha-1}{\alpha} + \frac{\alpha+1}{\alpha} Mat$

Proof. *Let $\mathcal{L}$ be a list with largest item φ. We use $W(x) = \frac{\alpha+1}{\alpha}x$ as a recurrent weighting function and consider $c(\mathcal{L}) = FF(\mathcal{L}) - W(\mathcal{L})$. First we will prove the lemma for $FF(\mathcal{L}) \leqslant 2$.*

1) $FF(\mathcal{L}) = 1 \;\Rightarrow\; Mat > \frac{1}{\alpha+1} \;\Rightarrow\; c < 1 - \frac{\alpha+1}{\alpha} \times \frac{1}{\alpha+1} = \frac{\alpha-1}{\alpha}$.

2) $FF(\mathcal{L}) = 2 \;\Rightarrow\; Mat > 1 \;\Rightarrow\; c < 2 - \frac{\alpha+1}{\alpha} = \frac{\alpha-1}{\alpha}$.

Now assume that $FF(\mathcal{L}) \geqslant 3$ and make the following observations. Let w_1 be the wastage in the first bin, and $W_{1,2}$ the weight of bins 1 and 2.

a) If $w_1 < \frac{1}{\alpha+1}$, then $W_1 = \frac{\alpha+1}{\alpha}(1 - w_1) > 1$.

b) If $w_1 \geqslant \frac{1}{\alpha+1}$, then the bin weight of the first two bins satisfies $W_1 + W_2 > 2$.

The second observation follows from: The smallest item in the 2^{nd} *bin,* s_2 *is strictly larger than the wastage in the first bin (use the invariant). So that* $s_2 > w_1 \geqslant \frac{1}{\alpha+1}$ *and since the items are no larger than* $1/\alpha$ *it follows that all items are* α*-items. If there are less than* α *items in the bin, it is filled to a level* $l_2 \leqslant (\alpha - 1) \times \varphi \leqslant 1 - 1/\alpha$ *and FF would have placed another item in the bin. So that there are exactly* α *items in the second bin and all of them are* α*-items. The total amount of material packed in the first two bins is thus* $l_1 + l_2 \geqslant 1 - w_1 + \alpha s_2 > 1 - w_1 + \alpha w_1 \geqslant 1 + \frac{\alpha-1}{\alpha+1}$*. So that the weight of these bins is* $W_1 + W_2 = \frac{\alpha+1}{\alpha}(l_1 + l_2) > \frac{\alpha+1}{\alpha}(1 + \frac{\alpha-1}{\alpha+1}) = 2$.

This means that if we have a list with $FF(\mathcal{L}) \geqslant 4$ *we can create a smaller list with larger c-value, by deleting either the first bin or the first two bins. Repeating this we end up with a list with either* $FF(\mathcal{L}) = 3$ *or* $FF(\mathcal{L}) = 2$*. For the latter the lemma has already been proven, so that only one case remains.*

3) $FF(\mathcal{L}) = 3 \;\Rightarrow\; Mat > 1 + \alpha \times \frac{1}{\alpha+1} \;\Rightarrow\; c < 3 - \frac{\alpha+1}{\alpha} - 1 = \frac{\alpha-1}{\alpha}$.

We can make the assumption $w_1 \leqslant \frac{1}{\alpha+1}$*, otherwise case (2) follows, and thus that the second bin contains exactly* α α*-items. For the first item placed in the third bin,* x *and the level of the first bin we have* $l_1 + x > 1$ *by the FF-rule. The total amount of material packed in the three bins is therefore at least* $1 + \alpha \times \frac{1}{\alpha+1}$*, and the lemma follows for this case.*
Ergo, for all cases we have $c \leqslant \frac{\alpha-1}{\alpha}$*, and this proves the lemma.*[2] □

6.9 Claim *The bound in lemma 6.8 is the best possible*

Proof. *Take a list with* $(k-1)\alpha + 1$ *items of size* $\frac{1}{\alpha+1} + \frac{\varepsilon}{k(\alpha+1)}$*. For* ε *sufficiently small, first-fit will pack this list into* k *bins. This gives* $FF = \frac{\alpha-1}{\alpha} - \frac{\varepsilon}{\alpha k}[(k-1)\alpha + 1] + \frac{\alpha+1}{\alpha} Mat$ *and since* $(k-1)\alpha + 1 \leqslant \alpha k$ *we have that for every* $\varepsilon > 0$ *there is a list such that* $FF \geqslant \frac{\alpha-1}{\alpha} - \varepsilon + \frac{\alpha+1}{\alpha} Mat$*. This proves the claim.* □

We can use lemma 6.8 to derive the following lemmas.

6.10 Lemma $\varphi \leqslant 1/\alpha$ *and* $Mat > \frac{1}{\alpha+1} \Rightarrow FF < \frac{\alpha-1}{\alpha} + \frac{\alpha+1}{\alpha} Mat$

Proof. *For* $Mat \leqslant 1$ *we have* $FF = 1$ *and the lemma is obvious. Now assume* $Mat > 1$ *and let* $\beta \geqslant \alpha$ *be such that* $\varphi \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$*. Applying lemma 6.8 gives* $FF < 2 + \frac{\beta+1}{\beta}(Mat - 1)$ *from which* $FF < 2 + \frac{\alpha+1}{\alpha}(Mat - 1)$ *follows.* □

[2]It appears that we can prove the lemma by reducing a list $\mathcal{L}$ to either case 1) or case 2) but this is not so. Although the list $\mathcal{L}$ contains an α-item, we cannot assume that this holds for the resulting list, after we have deleted the first [two] bin[s]. All α-items might have been in these bins, and we could not have applied the bound of case 1).

6.11 Lemma $\varphi \leqslant 1/\alpha \Rightarrow FF \leqslant \frac{\alpha-2}{\alpha} + \frac{\alpha+1}{\alpha} OPT$

Proof. *For $OPT = 1$ we have $FF = 1$ and the lemma is obvious. If $OPT \geqslant 2$, then (3.1) implies $Mat > 1$ and lemma 6.10 applies. Now replace Mat by OPT and we can tighten the '< 2' to '$\leqslant 2 - \frac{1}{\alpha}$', and the lemma follows.* □

The bound in lemma 6.11 is sharp for $\alpha \geqslant 2$. An instance with $FF = 3$ and $OPT = 2$ is easily constructed as

$$\mathcal{L} = \{\ \alpha,\ \underbrace{\alpha+1, \ldots, \alpha+1}_{\alpha}\ \underbrace{\alpha, \ldots, \alpha}_{\alpha}\ \} \text{ on bins of size } \alpha^2 + \alpha, \quad \alpha \geqslant 2 \tag{6.20}$$

Johnson,[37] gives constructions for a list with largest item $\varphi \leqslant 1/\alpha$ to achieve $FF = \lceil \frac{k(\alpha+1)-1}{\alpha} \rceil$ and $OPT = k$. Selecting $k = 2 + \alpha m$ yields $FF = 3 + (\alpha+1)m$ and $OPT = 2 + \alpha m$, and achieves the bound given in lemma 6.11. These constructions involve lists which contain items with sizes which are small perturbations of the unit fraction $\frac{1}{\alpha+1}$.

6.8 Comments

1) Lemmas 6.8, 6.10 and 6.11 also apply for $\alpha = 1$, in particular $FF \leqslant -1 + 2OPT$. Note that this bound is the next-fit bound (corollary E13, p. 248), and that this bound is as good as (6.2) for $OPT \leqslant 6$. Lemma 6.11 gives $FF \leqslant 3/2 OPT$ for lists with no 1-items. Combined with the instance in (6.20) this shows that the absolute worst-case performance ratio for FF, when restricted to lists on $\langle 0, \frac{1}{2}]$ is 1.5, that is $\forall \varphi \leqslant 1/2 \ R_{FF}(\varphi) = 1.5$ [3]

2) The proof of (6.1) in Johnson et al.,[37] and its subsequent refinement in Garey et al.[30] are based on the weighting functions W_2 and W_3 respectively.

$$W_2(x) = \begin{cases} \frac{6}{5}x, & 0 < x \leqslant \frac{1}{6} \\ \frac{9}{5}x - \frac{1}{10}, & \frac{1}{6} < x \leqslant \frac{1}{3} \\ \frac{6}{5}x + \frac{1}{10}, & \frac{1}{3} < x \leqslant \frac{1}{2} \\ 1, & \frac{1}{2} < x \leqslant 1 \end{cases} \quad \text{and} \quad W_3(x) = \begin{cases} \frac{6}{5}x, & 0 < x \leqslant \frac{1}{6} \\ \frac{9}{5}x - \frac{1}{10}, & \frac{1}{6} < x \leqslant \frac{1}{3} \\ \frac{6}{5}x + \frac{1}{10}, & \frac{1}{3} < x \leqslant \frac{1}{2} \\ \frac{6}{5}x + \frac{4}{10}, & \frac{1}{2} < x \leqslant 1 \end{cases} \tag{6.21}$$

Both these weighting functions are recurrent, with respect to the FF-heuristic. To prove this it is sufficient to show that $W_{2,3} \geqslant W(x)$ and use the fact that $W(x)$ as defined in (6.14) is recurrent.

[3] This extends the results for the general case; viz. $\exists \mathcal{L}\ 1.7 \leqslant \frac{FF(\mathcal{L})}{OPT(\mathcal{L})}$ and $\forall \mathcal{L}\ \frac{FF(\mathcal{L})}{OPT(\mathcal{L})} \leqslant 1.75$. The lower bound follows from the instance in Johnson[38] or the instance in diagram 6.1. The upper bound was proved by Simchi-Levi.[64]

3) The instances,[37] that were used to construct a lower bound for first-fit have $FF = 17k$ and $OPT = 10k + 1$. They further have $Mat = 10k + \left(2(18^k - 1) + 10k\right)\delta$ and a smallest item $s = \frac{1}{6} - 13 \times 18^{k-1}\delta$, where $\delta \ll 18^{-k}$, which gives $W_l = OPT - Mat = 1 - \left(2(18^k - 1) + 10k\right)\delta$. For δ sufficiently small we have $W_l > 1 - s$, and by the proof of lemma 5.14 we can create a list with $FF = 17k + 1$ and $OPT = 10k + 1$.

   This implies that $\forall N \; \exists\{\mathcal{L} \mid OPT \geqslant N\} \; FF \geqslant -0.7 + 1.7OPT$, which improves slightly on the lower bound by Johnson et al.,[37] which is also quoted in Garey and Johnson [31, p. 125]

4) Johnson[37] also raises the question of what can be said about $FF(\mathcal{L})/OPT(\mathcal{L})$ for $OPT(\mathcal{L}) \to \infty$. We can use lemma 5.15 to partially answer the question;

$$\forall \mathcal{L} \; \lim_{k\to\infty} \frac{FF(k\mathcal{L})}{OPT(k\mathcal{L})} < 1.7 \tag{6.22}$$

   Note that this leaves open the question of the absolute worst-case ratio. It may well be that there is an instance for which $FF/OPT > 1.7$ holds.

   Moreover, for any fixed list $\mathcal{L}$, or (integer) lists with a maximum bin-size (viz. 16 or 32-bit integers in an encoding of the list) we have that the difference $FF(\mathcal{L}) - 1.7OPT(\mathcal{L})$ cannot be lower bounded by a constant.

5) The Next-k-Fit (NkF) heuristic is a fixed space variant of first-fit. In this heuristic we have at most k active bins. If there are less than k bins active it works exactly as first-fit. If there are k bins active and an item cannot be placed, the first bin is closed and a new bin is opened in which the item is placed. Note that N1F $\cong$ NF and N∞F $\cong$ FF.

   Johnson[38] notes that for $k \geqslant 2$ and lists with largest item $\varphi \leqslant 1/2$ the heuristics NkF and FF have the same asymptotic ratio. We note that lemma 6.8, and with it lemmas 6.10 and 6.11, also hold for NkF; one need only replace '*FF*' by '*NkF*' in its proof. This implies

$$k \geqslant 2 \text{ and } \varphi \in \left\langle\tfrac{1}{\alpha+1}, \tfrac{1}{\alpha}\right] \;\Rightarrow\; NkF < \tfrac{\alpha-1}{\alpha} + \tfrac{\alpha+1}{\alpha} Mat \tag{6.23}$$

$$k \geqslant 2 \text{ and } \varphi \leqslant 1/\alpha \;\Rightarrow\; NkF \leqslant \tfrac{\alpha-2}{\alpha} + \tfrac{\alpha+1}{\alpha} OPT \tag{6.24}$$

   This bound is sharp for $\alpha \geqslant 2$ (use the worst-case instances for FF).

   Note that for $k \geqslant 2$, NkF has the invariant $\forall 0 < j - i < k \; w_i < s_j$, which is similar to the invariant for FF. The N1F-invariant (appendix E.4) however, is different from the FF-invariant.

Chapter 7

Next-Fit Decreasing Heuristic

7.1 Introduction

The *Next-Fit Decreasing (NFD)* heuristic takes a list $\mathcal{L}$, sorts it into non-increasing order of item size and packs this list according to the next-fit rule. The *Next-Fit (NF)* rule places each item, in succession, in the current bin. When an item cannot be placed the current bin is closed, and a new bin is opened in which this item is placed. As an illustration we give the following example.

List $\mathcal{L} = \{ \underbrace{21,\ldots,21}_{6},\ \underbrace{14,\ldots,14}_{6},\ \underbrace{6,\ldots,6}_{6} \}$ on bins of size 41.

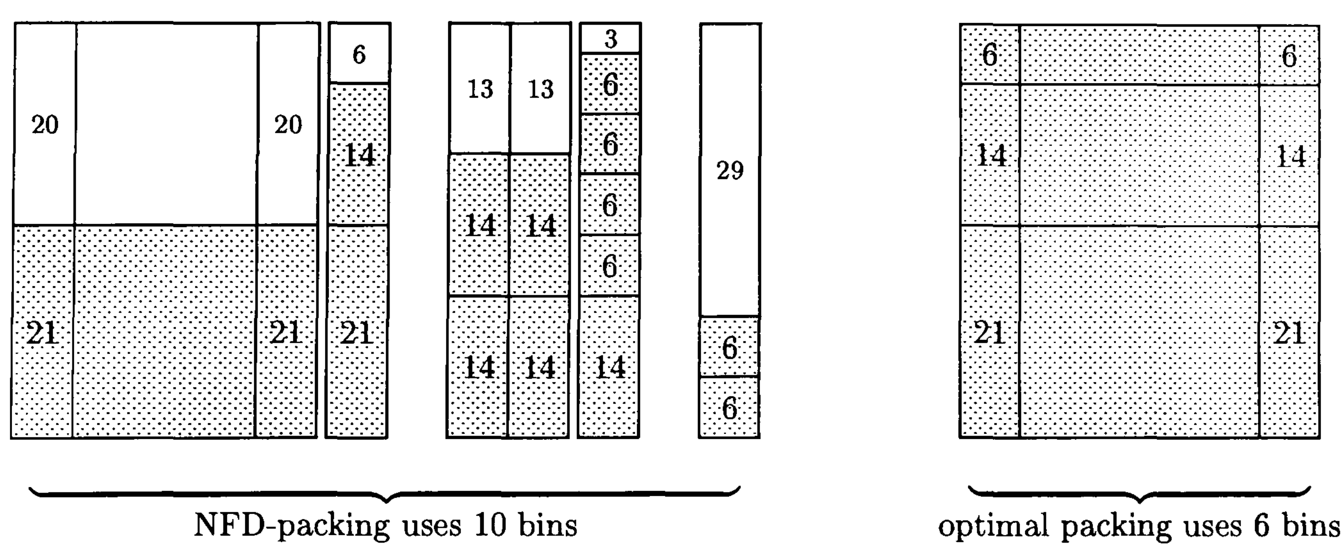


Diagram 7.1. An example of packing by next-fit decreasing.

The next-fit decreasing heuristic was studied by Baker and Coffman.[4] They showed that its asymptotic ratio, $R^{\infty}_{NFD} = 1.69103...$ and derived the following bound

$$NFD \leqslant 3 + R^{\infty}_{NFD} \times OPT \tag{7.1}$$

Their derivation is based upon a weighting function which satisfies properties (5.2) and (5.3). The rationale for the weighting function they use, replicated for reference in (7.25), is not given except that it is a function which satisfies the aforementioned properties.

For lists with items restricted to the interval $\langle 0, 1/\alpha]$ they give the bound $NFD \leqslant 3 + R^{\infty}_{NFD}(1/\alpha) \times OPT$, where the parametric asymptotic ratio, $R^{\infty}_{NFD}(1/\alpha)$ is given by (7.14). They further establish that these ratios are tight by constructing instances for which $NFD(\mathcal{L})/OPT(\mathcal{L})$ approximates the ratio as closely as desired.

In this chapter we will present a proof, based on the concepts developed in chapter 5, to sharpen the NFD-bound to

$$NFD < 1.14793 + R^{\infty}_{NFD} \times CSP_R \tag{7.2}$$

where CSP_R is the value of the LP-relaxation of the bin-packing problem, formulated as a cutting stock problem. For restricted lists we will show that

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha} \right] \quad \Rightarrow \quad NFD < 1 + \tfrac{3}{(\alpha+1)(\alpha+2)} \;+\; R^{\infty}_{NFD}(1/\alpha) \times CSP_R \tag{7.3}$$

where φ is the largest item in the list and $\alpha = \lfloor 1/\varphi \rfloor$. We give simpler and aesthetically more pleasing examples to show that the [parametric] ratios are the best possible. These instances are based directly upon the properties of the doubly exponential sequence that defines these ratios.

We start the analysis of NFD by determining the structure of recurrent bins and show how this naturally leads to a weighting function. This translates into a 'ratio'-problem which is solved in appendix B.1. The solution of this directly yields the sequence of numbers $1, 2, 6, 42, 1806, \ldots$ (and similar sequencies for the parametric case) on which the weighting function to derive (7.1) is based.

Notation By s and l we denote the smallest and largest item, respectively and by w the wastage in a bin. We will use the shorthand N to denote the number of bins a list packs into; $N = NFD(\mathcal{L})$. As usual, W_j and W_{bin} denote the weight of a bin.

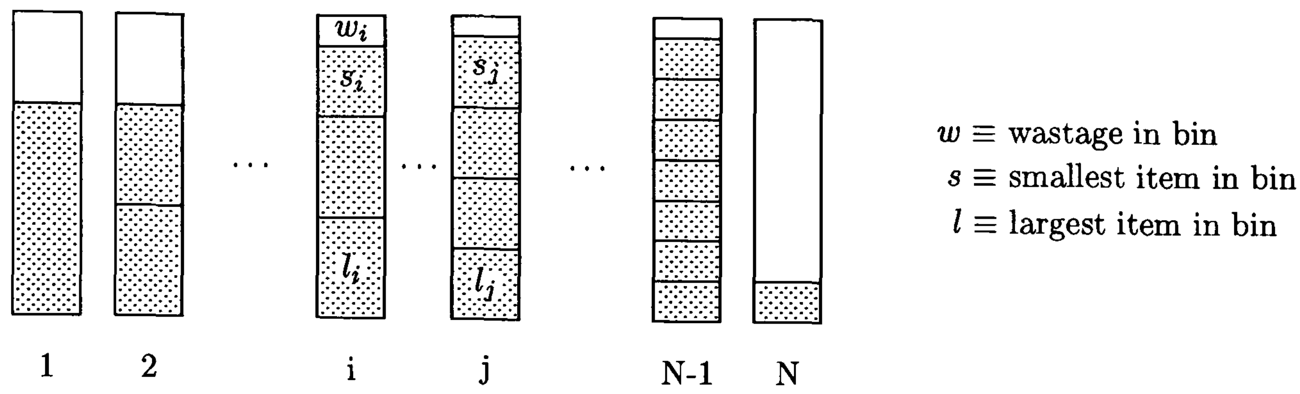


Diagram 7.2. Minimal NFD-configuration.

7.2 Canonical form

Suppose that we have packed a list $\mathcal{L} = \{x_1, \ldots, x_n\}$ using the NFD-heuristic. We may assume wlog that the list is in non-increasing order of size; $x_1 \geqslant \cdots \geqslant x_n$. Additionally, for a minimal list, we may assume that the last bin is a singleton bin. The NFD-rule leads directly to the following invariant

$$\textbf{NFD-invariant:} \quad \forall 1 < j \leqslant N \quad w_{j-1} < l_j \tag{7.4}$$

As the items are packed in non-increasing order we have $s_{j-1} \geqslant l_j$, and the following invariant

$$\textbf{NFD-invariant:} \quad \forall 1 \leqslant j < N \quad w_j < s_j \tag{7.5}$$

7.3 Recurrent patterns

Any pattern or bin that occurs more than once in a realisation of NFD must consist of items that are all of the same size. This follows directly from the invariant and the fact that the list $\mathcal{L}$ is sorted into non-increasing order. For instance, take bin i and bin j (diagram 7.2); by the NFD-rule we have $l_i \geqslant s_i \geqslant l_j \geqslant s_j$. If these bins are identical then $l_i = l_j$ and $s_i = s_j$, so that all the sizes in these bins are the same. Moreover, if x is the size of the items in such a bin, it must contain the maximum number of items of size x that will fit. This leads to the following characterisation of a recurrent bin.

7.1 Definition ***(NFD-recurrent)*** *A bin, in which all the items are of the same size x, and which contains exactly $\lfloor 1/x \rfloor$ of such items, is said to be recurrent.*

7.2 Definition ***(NFD-recurrent weighting function)*** *Any non-decreasing function that satisfies $W(x) \geqslant 1/\lfloor 1/x \rfloor$*

7.4 Weighting function

The structure of a recurrent bin leads directly to the following (minimal) weighting function[1]

$$W(x) = \frac{1}{\lfloor 1/x \rfloor}, \tag{7.6}$$

or alternatively

$$W(x) = \frac{1}{i}, \quad \text{for } x \in \left\langle \tfrac{1}{i+1}, \tfrac{1}{i} \right] . \tag{7.7}$$

7.5 General bound

The relatively simple structure of the weighting function, combined with the monotonicity of the NFD-heuristic allows the derivation of a priori bounds in terms of the number of i-items. To this end we derive the following lemmas. The weighting function is assumed to be NFD-recurrent.

7.3 Lemma *If all the items in a bin, excluding the last [singleton] bin, are i-items then* $W_{\text{bin}} \geqslant 1$

Proof. *Say there are n i-items in a bin. Then* $n \times 1/i \geqslant \sum x_i = 1 - w > 1 - s \geqslant 1 - 1/i \Rightarrow n > i - 1$. *Since there can be at most i i-items in a bin it follows that* $n = i$. *The weight of an i-item is at least* $1/i$, *so that the bin weight is at least one.* □

7.4 Lemma *If there are* $i < j < N$ *such that* $W_i < 1$ *and* $W_j < 1$ *then* $\lfloor 1/s_i \rfloor < \lfloor 1/s_j \rfloor$

Proof. *Take diagram 7.2 and assume that the weight of both bin i and bin j is strictly less than one. By the NFD-rule we have* $s_i \geqslant s_j$, *so that* $\lfloor 1/s_i \rfloor \leqslant \lfloor 1/s_j \rfloor$ *holds. Now assume that equality holds and denote* $\alpha = \lfloor 1/s_i \rfloor = \lfloor 1/s_j \rfloor$. *We have* $s_i \geqslant l_j \geqslant s_j$, *so that* $\lfloor 1/s_i \rfloor \leqslant \lfloor 1/l_j \rfloor \leqslant \lfloor 1/s_j \rfloor$ *implies* $\lfloor 1/l_j \rfloor = \alpha$. *This means that all the items in bin j are α-items and therefore by lemma 7.3 that* $W_j \geqslant 1$. *This leads to a contradiction and therefore* $\lfloor 1/s_i \rfloor < \lfloor 1/s_j \rfloor$ *must hold and proves the lemma.* □

As a consequence of lemma 7.4 we have the following corollary.

7.5 Corollary *Under a recurrent weighting function we have that for every i-interval there can be at most one bin, among the first* $N - 1$, *with bin weight strictly less than one.*

[1] In fact the structure of the recurrent patterns directly gives a formulation for the ratio problem (5.24). The recurrency gives the requirement $u_i \geqslant 1/\alpha_i$, so that the ratio problem becomes: $\max\{\sum a_i/\alpha_i \mid \mathbf{a} \in \mathcal{A}\}$.

7.6 Lemma *Every bin with smallest item s, excluding the last (singleton) bin, has weight* $W_{\text{bin}} > 1 - s \geqslant 1 - 1/\lfloor 1/s \rfloor$

Proof. *Consider the following program for a NFD-recurrent weighting function.*

$$W_{\text{bin}} \geqslant \boxed{\begin{array}{ll} \text{Min} & \sum W(x_i) \\ \text{st} & \sum x_i = 1 - w \\ & x_i > 0 \end{array}} \geqslant \boxed{\begin{array}{ll} \text{Min} & \sum \frac{a_i}{i} \\ \text{st} & \sum \frac{a_i}{i} > 1 - s \\ & a_i \in \mathrm{N} \end{array}} \qquad (7.8)$$

This immediately proves the lemma, since $W_{\text{bin}} = \sum W(x_i) \geqslant \sum a_i/i > 1 - s$*. To arrive at the second program we have used the invariant* $w < s$ *and assumed that there are* a_i *i-items.* □

7.7 Lemma *If* $\mathcal{L}$ *is a list with* $f_i \geqslant 1$ *items in the interval* $\langle \frac{1}{\alpha_i+1}, \frac{1}{\alpha_i}]$*, where* $1 \leqslant i \leqslant m$ *and* $1 \leqslant \alpha_1 < \cdots < \alpha_m$*, then* $NFD(\mathcal{L}) < 1 + \sum_{i=2}^{m-1} 1/\alpha_i + \sum_{i=1}^{m} f_i/\alpha_i$

Proof. *We take weighting function (7.7) and consider* $c(\mathcal{L}) = NFD(\mathcal{L}) - W(\mathcal{L}) = \sum(1 - W_j)$*, where the summation runs from 1 to* $NFD(\mathcal{L})$*. We derive a bound for* $c(\mathcal{L})$ *by a case analysis.*

1) If $m = 1$ *then* $NFD = \lceil f_1/\alpha_1 \rceil$ *and thus* $c \leqslant 1 - 1/\alpha_1$*. Note that NFD is optimal for this case.*

Now assume $m \geqslant 2$*, and define* δ_i *as the number of bins, not counting the last [singleton] bin, with an* α_i*-item as smallest item and bin weight strictly less than one. From corollary 7.5 it follows that for every* α_i*-interval there can be at most one bin with bin weight strictly less than one, so that* $\delta_i \in \{0, 1\}$*. Furthermore, there cannot be a bin with an* α_1*-item as smallest item and bin weight* < 1*, so that* $\delta_1 = 0$*.*

2) If $\forall i\ \delta_i = 0$ *then* $c(\mathcal{L}) = \sum(1 - W_j) \leqslant 1 - W_{\text{singleton}} \leqslant 1 - 1/\alpha_m$*, and the lemma holds.*

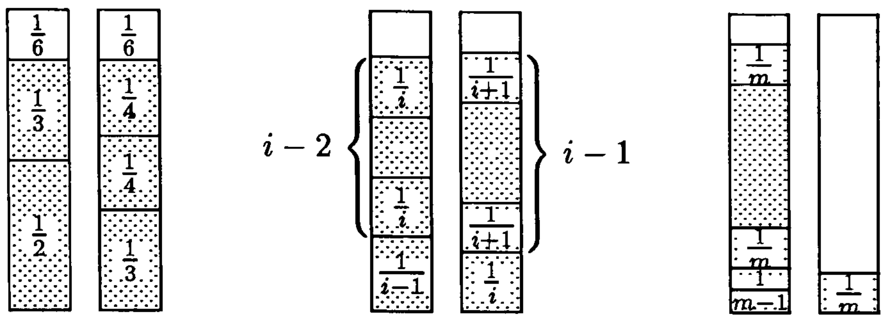
3) If $\exists i\ \delta_i = 1$ *then, by corollary 7.5 and lemma 7.6*

$$c(\mathcal{L}) \leqslant \sum_{j|W_j<1} (1 - W_j) < \sum_{i=1}^{m} \delta_i/\alpha_i + 1 - W_{\text{singleton}} \leqslant \sum_{i=2}^{m} \delta_i/\alpha_i + 1 - 1/\alpha_m \leqslant 1 + \sum_{i=2}^{m-1} 1/\alpha_i$$

and completes the proof of the lemma. □

To show that the bound in lemma 7.7 is a reasonably tight bound we have given an example in diagram 7.3 for the special case when $\forall i\ \alpha_i = i + 1$ (see also comment 5, page 75). Note that the monotonicity of NFD and the fact that the weighting function is constant on an i-interval implies that we may assume that lists that maximise $NFD - \sum f_i/i$ are harmonic; that is all items in the list are unit fractions.

List $\mathcal{L} = \{ \underbrace{\tfrac{1}{i}, \ldots, \tfrac{1}{i}}_{i-1} \}, \; i = 2, \ldots, m$ on bins of size 1.



$$NFD(\mathcal{L}) = m - 1$$

$$W(\mathcal{L}) = \sum_{i=2}^{m} \tfrac{i-1}{i}$$

$$c(\mathcal{L}) = \sum_{i=2}^{m} \tfrac{1}{i}$$

Diagram 7.3. An example of a packing by next-fit decreasing.

A bound slightly less tight, but more general can be obtained as follows. For brevity denote $\alpha = \alpha_1$ and $\beta = \alpha_m$, so that $\alpha_i \geqslant \alpha + i - 1$ and in particular $\beta \geqslant \alpha + m - 1$. We can relax the bound in lemma 7.7 using $\alpha_i \geqslant \alpha + i - 1$ and a simple area argument.

$$\sum_{i=2}^{m-1} \frac{1}{\alpha_i} \leqslant \sum_{i=2}^{m-1} \frac{1}{\alpha+i-1} = \sum_{i=\alpha+1}^{\alpha+m-2} \frac{1}{i} \leqslant \int_{\alpha}^{\alpha+m-2} \frac{1}{x} dx = \ln\left(\frac{\alpha+m-2}{\alpha}\right), \qquad \text{for } m \geqslant 2 \tag{7.9}$$

Denote by n_i the number of items in the interval $\langle \frac{1}{i+1}, \frac{1}{i}]$ and recall that $m = \sum_i 1_{\{n_i>0\}}$. This gives the following corollaries.[2]

7.8 Corollary $NFD < 1 + \ln(1 + \frac{m-1}{\alpha}) + \sum n_i/i$

and since $m \leqslant \beta + 1 - \alpha$

7.9 Corollary $\mathcal{L} \subset \langle \frac{1}{\beta+1}, \frac{1}{\alpha}] \;\Rightarrow\; NFD < 1 + \ln(\beta/\alpha) + \sum n_i/i$

[2]Note that, since $m \leqslant n$, the bound in lemma 7.7 and corollary 7.8 improve upon the upper bound implied by $|NFD - \sum n_i/i| \leqslant 1 + \ln(n)$, given by Rhee.[56]

7.6 Asymptotic ratio

To determine the asymptotic ratio of NFD we solve the ratio program with W defined as in (7.7),

$$r = \boxed{\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i \leqslant 1 \\ & x_i \geqslant 0 \end{array}} \tag{7.10}$$

By corollary A2 we may assume that the x_i are rationals. Say $x_i = d_i/L$ with $d_i, L \in \mathbb{N}^+$ and define $\alpha_i = \lfloor 1/x_i \rfloor = (L \text{ div } d_i)$. We then have

$$W(x_i) = \frac{1}{\alpha_i}, \quad \text{for } x_i \in \langle \tfrac{1}{\alpha_i+1}, \tfrac{1}{\alpha_i}] \quad \Rightarrow \quad W(d_i/L) = \frac{1}{\alpha_i}, \quad \text{for } d_i \in [\tfrac{L+1}{\alpha_i+1}, \tfrac{L}{\alpha_i}] \tag{7.11}$$

since $d_i/L > \frac{1}{\alpha_i+1}$ implies $d_i \geqslant \frac{L+1}{\alpha_i+1}$. If we now substitute $x_i = d_i/L$, use the fact that $d_i \geqslant \frac{L+1}{\alpha_i+1}$ and group sizes with equal weight we get an upper bound for (7.10) from the following program.

$$r \leqslant \boxed{\begin{array}{ll} \text{Max} & \sum_{i=1}^{m} \frac{a_i}{\alpha_i} \\ \text{st} & \sum_{i=1}^{m} \frac{L+1}{\alpha_i+1} a_i \leqslant L \\ & a_i \in \mathbb{N} \end{array}} \tag{7.12}$$

Let x_1 be the largest item in the list and $\alpha = \alpha_1 = \lfloor 1/x_1 \rfloor$. We now scale the constraint by dividing by $(L+1)$, extend the summation, let $L \to \infty$ and arrive at the following knapsack problem;

$$r(\alpha) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i+1} < 1 \\ & a_i \in \mathbb{N} \end{array}} \tag{7.13}$$

This is a problem which is studied in appendix B.1 and yields the solution-value [3]

$$R^{\infty}_{NFD}(1/\alpha) = 1 + \sum_{i=1}^{\infty} \frac{1}{b_i}, \quad \text{where } b_1 = \alpha + 1 \text{ and } b_{i+1} = b_i(b_i+1) \tag{7.14}$$

in particular

$$R^{\infty}_{NFD}(1) = 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{42} + \frac{1}{1806} + \frac{1}{3263442} + \cdots \approx 1.691030206... \tag{7.15}$$

[3]More detail on the sequence $\{b_i(\alpha)\}$, $r(\alpha)$ and its rate of convergence can be found in appendix B.2.

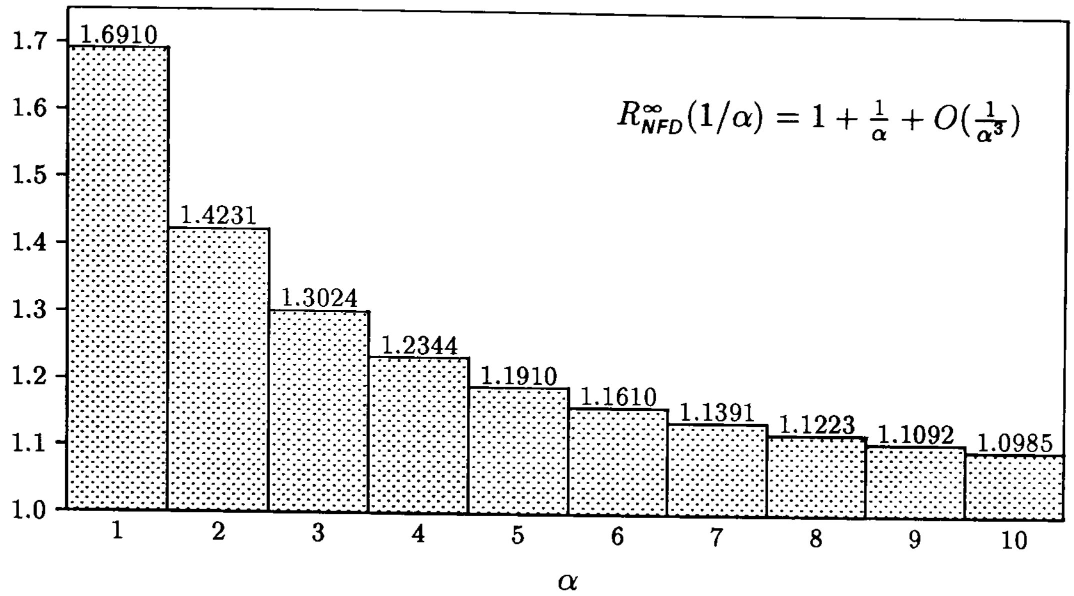


Diagram 7.4. NFD-asymptotic ratios for $\mathcal{L} \subset \langle 0, \frac{1}{\alpha}]$.

Bounds for the parametric asymptotic ratio can be obtained from corollaries B3–B7 and lemma B13.

$$1 + \tfrac{1}{\alpha+1} < 1 + \tfrac{1}{\alpha+1} + \tfrac{1}{(\alpha+1)(\alpha+2)} < R^{\infty}_{NFD}(1/\alpha) < 1 + \tfrac{1}{\alpha+1} + \tfrac{1}{(\alpha+1)(\alpha+2)-1} < 1 + \tfrac{1}{\alpha} \qquad (7.16)$$

We can characterise the asymptotic behaviour by the following corollary,[4] where the last expression follows from $1 + \frac{1}{\alpha} - R^{\infty}_{NFD}(1/\alpha) < \frac{2}{\alpha(\alpha+1)(\alpha+2)}$.

7.10 Corollary $R^{\infty}_{NFD}(1/\alpha) = 1 + \frac{1}{\alpha+1} + O(\alpha^{-2})$ *and* $R^{\infty}_{NFD}(1/\alpha) = 1 + \frac{1}{\alpha} + O(\alpha^{-3})$

7.7 Instances

To prove that (7.14) is the asymptotic ratio we construct lists that approximate this ratio as closely as desired. These constructions improve upon the ones given by Baker and Coffman.[4]

For a given α choose a positive integer N, and calculate $b_1, \ldots, b_N$ as in (7.14). Now construct the following list (note that by property B10 all the sizes in $\mathcal{L}$ are integer).

$$\mathcal{L} = \Bigg\{ \underbrace{\frac{b_N}{\alpha+1}, \ldots, \frac{b_N}{\alpha+1}}_{\alpha}, \underbrace{\frac{b_N}{b_1+1}, \ldots, \frac{b_N}{b_n+1}, \ldots, \frac{b_N}{b_{N-1}+1}}_{\text{1 item for each } n} \Bigg\} \text{ on bins of size } b_N - 1 \qquad (7.17)$$

We can pack this list into exactly one bin with zero wastage, since by property B12 we have;

$$Mat(\mathcal{L}) = \frac{b_N}{b_N - 1}\left(\frac{\alpha}{\alpha+1} + \sum_{n=1}^{N-1} \frac{1}{b_n+1}\right) = \frac{b_N}{b_N-1}\left(1 - \frac{1}{b_N}\right) = 1 \;\Rightarrow\; OPT(\mathcal{L}) = 1 \qquad (7.18)$$

[4] This refines the statement[4] that $R^{\infty}_{NFD}(1/\alpha)$ approaches 1 with increasing α as $(\alpha+2)/(\alpha+1)$.

Now choose k as a multiple of b_{N-1} and construct the list $k\mathcal{L}$. This list has a number of $k\alpha$ of α-items and k of b_n-items ($1 \leqslant n \leqslant N-1$). Then, by corollary E8 and property B9 we have;

$$NFD(k\mathcal{L}) = \frac{k\alpha}{\alpha} + \sum_{n=1}^{N-1} \frac{k}{b_n} = k\Big(1 + \sum_{n=1}^{N-1} \frac{1}{b_n}\Big) \tag{7.19}$$

Since $OPT(k\mathcal{L}) = kOPT(\mathcal{L})$ we now have a list $k\mathcal{L}$ such that $NFD/OPT = 1 + \sum_{n=1}^{N-1} \frac{1}{b_n}$ for any value of N and k.[5] This means that we can approximate the [parametric] asymptotic ratio as closely as desired by choosing N accordingly.

From the tables B.1 and B.2 we see that for all values of α, a choice of $N = 3$ gives instances that give a close approximation[6] of $R^{\infty}_{NFD}(1/\alpha)$, with sizes/numbers that are 'reasonably' small. An example using the above construction for $\alpha = 1$ and $N = 3$ is given in diagram 7.1.

7.8 Constant

Let $\mathcal{L}$ be a list with n items, largest item $\varphi \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ and smallest item x_n. Now determine the positive integer N, such that $\frac{1}{b_N(\alpha)} < x_n \leqslant \frac{1}{b_{N-1}(\alpha)}$, with $b_n(\alpha)$ the sequence as defined by (7.14). For the remainder of this section we use b_n to denote $b_n(\alpha)$. Consider $c(\alpha)$, which is defined as

$$c(\alpha) = \max_{\{\mathcal{L} | \frac{1}{\alpha+1} < \varphi(\mathcal{L}) \leqslant \frac{1}{\alpha}\}} c(\mathcal{L}) = NFD(\mathcal{L}) - W(\mathcal{L}) \tag{7.20}$$

with the weighting function $W(x)$ defined as

$$W(x) = \begin{cases} \rho_0 x, & x \in \langle \frac{1}{\rho_0 \alpha}, \frac{1}{\alpha}] \\ \frac{1}{\alpha}, & x \in \langle \frac{1}{\alpha+1}, \frac{1}{\rho_0 \alpha}] \\ \rho_n x, \quad 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{\rho_n b_n}, \frac{1}{b_n}] \\ \frac{1}{b_n}, \quad 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{b_n+1}, \frac{1}{\rho_n b_n}] \\ \rho_n x, \quad 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{b_{n+1}}, \frac{1}{b_n+1}] \end{cases} \tag{7.21}$$

and $\rho_n = \frac{b_n+2}{b_n+1}$ for $n \in \mathbb{N}$. We will show that

$$c(\alpha) < \begin{cases} 1.14793, & \alpha = 1 \\ 1 + \frac{3}{(\alpha+1)(\alpha+2)}, & \alpha \geqslant 2 \end{cases} \tag{7.22}$$

This weighting function is chosen as an extension and strengthening of (7.7). Before we derive bounds for $c(\mathcal{L})$ we make the following observation and eliminate some trivial cases.

[5] Obviously, the same applies for CSP_R and Mat, since $OPT(\mathcal{L}) = CSP_R(\mathcal{L}) = Mat(\mathcal{L})$.

[6] Actually, the error is strictly less than $(\alpha+1)^{-4}$; use bound (B7).

7.11 Observation $\rho_0 > \rho_1 > \cdots > \rho_n$

7.12 Assumption *$N \geqslant 2$ and $NFD \geqslant 2$*

Proof. *For $N = 1$ and f items in the interval $\langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$, we have $NFD = \lceil \frac{f}{\alpha} \rceil$, $\sum W(x_i) \geqslant \frac{f}{\alpha}$ and thus $c(\mathcal{L}) \leqslant 1 - \frac{1}{\alpha}$. For $NFD = 1$, we have $c(\mathcal{L}) \leqslant 1 - W(x_n) < 1$.*

For both these trivial cases (7.22) holds and the assumption follows. □

7.13 Assumption *$\mathcal{L}$ contains at least one and at most α α-items.*

Proof. *There is at least one by definition of α. If there are more, then the first bin must contain exactly α α-items and has a bin weight of at least 1. Therefore the list $\mathcal{L}'$, created by deleting the first α items from $\mathcal{L}$ has $c(\mathcal{L}') \geqslant c(\mathcal{L})$.* □

7.14 Assumption *The last bin is a singleton bin.*

Proof. *If there is more than one item in the last bin, we can delete all but the first item placed there. This will give a list which will pack in the same number of bins but has less total weight.* □

7.15 Assumption *$\mathcal{L}$ does not contain items in the intervals $\langle \frac{1}{b_n+1}, \frac{1}{\rho_n b_n} \rangle$.*

Proof. *Since the weighting function, W is constant on the interval $\langle \frac{1}{b_n+1}, \frac{1}{\rho_n b_n}]$ we can increase any size in this interval to a size $\frac{1}{\rho_n b_n}$. This does not increase $\sum_i W(x_i)$ nor decrease NFD, by virtue of its monotonicity, and therefore does not decrease $c(\mathcal{L})$.* □

A direct consequence of assumption 7.15 is the following corollary.

7.16 Corollary $x \in \langle \frac{1}{b_{n+1}}, \frac{1}{b_n}] \;\Rightarrow\; W(x) = \rho_n x$

7.17 Corollary $W_{\text{singleton}} = W(x_n) \;>\; \rho_{N-1}\frac{1}{b_N} \;>\; \left(\frac{1}{b_N}\right)^2 + \frac{1}{b_N}$.

A bound for $c(\mathcal{L})$ follows from the bins with weight < 1 as $c(\mathcal{L}) \leqslant \sum_{j|W_j<1} [1 - W_j]$. We now determine the structure of bins with weight strictly less than one. First define a transition bin (T-bin), similar to Baker[4] as

7.18 Definition ***(Transition bin)*** *Any bin, except the last bin, which has a smallest item x and a largest item y such that $y \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ and $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$, where $\beta \geqslant \alpha + 1$ is called a transition bin. Note that the last bin, assumed to be a singleton bin, is not a transition bin.*

By lemma 7.3 we have that any bin with weight < 1 must be a transition bin. We now condition on the size of the smallest item in a T-bin to derive a lower bound for its bin weight.

7.19 Lemma *If the smallest item in a bin, $x \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}] \subset \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$ then $W_{\text{bin}} > \frac{\alpha-1}{\alpha}\rho_n$*

Proof. *$W_{\text{bin}} = \sum W(x_i) \geqslant \sum \min_i \left(\frac{W(x_i)}{x_i}\right) x_i$. Now use corollary 7.16 and the NFD-invariant to give $W_{\text{bin}} > \rho_n \sum x_i > \rho_n(1-x) \geqslant \rho_n(1-1/\alpha)$.* □

As a direct consequence of this lemma we have the following corollaries.

7.20 Corollary ***(Bin weights of T-bins with smallest item x)***

a) $x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$ *and $n \in \mathbb{N}$* $\Rightarrow W_{\text{bin}} > 1 - \frac{2}{b_n(b_n+1)}$

b) $x \in \langle \frac{1}{b_n+2}, \frac{1}{b_n+1}]$ *and $n \in \mathbb{N}^+$* $\Rightarrow W_{\text{bin}} > 1 - \frac{1}{(b_n+1)^2}$

c) $x \in \langle \frac{1}{b_{n+1}}, \frac{1}{b_n+2}]$ *and $n \in \mathbb{N}^+$* $\Rightarrow W_{\text{bin}} > 1$

We can sharpen the first case of corollary 7.20 by the following lemma.

7.21 Lemma *If $x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$ is the smallest item in a transition bin and $n \geqslant 2$, then a lower bound for its bin weight is given by $W_{\text{bin}} > 1 + \frac{1}{b_{n+1}}\left[\frac{b_{n-1}^2-2b_{n-1}-2}{b_{n-1}+1}\right]$*

Proof. *Since it is a transition bin, the largest item it contains, $x_1 = y$ must satisfy $y > \frac{1}{b_n}$, and its weight $W(y) \geqslant \rho_{n-1}y$ by corollary 7.16 and observation 7.11. All other items have weight $W(x_i) \geqslant \rho_n x_i$. From the invariant it follows that $\sum x_i > 1-x$. Combining these gives*

$$W_{\text{bin}} = \sum W(x_i) = W(y) + \sum_{i\neq 1} W(x_i) \geqslant \rho_{n-1}y + \rho_n\left(\sum x_i - y\right) > \rho_{n-1}y + \rho_n(1-x-y)$$

This expression is minimised for y minimal, since $\rho_{n-1} > \rho_n$, and for x maximal. This lower bounds the weight as $W_{\text{bin}} > (\rho_{n-1}-\rho_n)\frac{1}{b_n} + \rho_n\left(1-\frac{1}{b_n}\right)$. Substituting the values for ρ_n and ρ_{n-1} gives:

$$W_{\text{bin}} > \left(\frac{b_{n-1}+2}{b_{n-1}+1} - \frac{b_n+2}{b_n+1}\right)\frac{1}{b_n} + \frac{b_n+2}{b_n+1}\left(\frac{b_n-1}{b_n}\right) = \cdots = 1 + \frac{1}{b_{n+1}}\left[\frac{b_n-3b_{n-1}-2}{b_{n-1}+1}\right]$$

Since $n \geqslant 2$ we can use the recurrency $b_n \equiv b_{n-1}(b_{n-1}+1)$ and substitution yields the lemma. □

Note that the expression in square brackets in lemma 7.21 is positive for $b_{n-1} \geqslant 3$ which implies the following corollary.

7.22 Corollary ***(Bin weight of T-bins with smallest item $x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$)***

1) $\alpha = 1$ *and* $n = 1 \Rightarrow W_{\text{bin}} \geqslant 1.5$

2) $\alpha = 1$ *and* $n = 2 \Rightarrow W_{\text{bin}} > \frac{62}{63}$ *(substitute $b_1 = 2, b_2 = 6, b_3 = 42$).*

3) $\alpha = 1$ *and* $n \geqslant 3 \;\Rightarrow\; W_{\text{bin}} > 1$ *(since* $b_2 = 6 \geqslant 3$ *and* b_n *is increasing with* n*).*

4) $\alpha \geqslant 2$ *and* $n = 1 \;\Rightarrow\; W_{\text{bin}} > 1 - \frac{2}{b_2}$ *(use corollary 7.20)*

5) $\alpha \geqslant 2$ *and* $n \geqslant 2 \;\Rightarrow\; W_{\text{bin}} > 1$ *(since* $b_1(\alpha) = \alpha + 1 \geqslant 3$ *and* b_n *is increasing with* n*).*

We now prove (7.22). For every i-interval we can have at most one bin with weight < 1, which combined with corollaries 7.20 and 7.22 gives the following bound for $c(1)$.

$$c(1) \;<\; 1 - \frac{62}{63} \;+\; \sum_{n=1}^{N-1} \left(\frac{1}{b_n + 1}\right)^2 \;+\; (1 - W_{\text{singleton}})$$

The first term corresponds to the T-bins with smallest item in $\langle \frac{1}{b_n+1}, \frac{1}{b_n}]$ and the summation to T-bins with smallest item in $\langle \frac{1}{b_n+2}, \frac{1}{b_n+1}]$. Now use lemma B14 and corollary 7.17.

$$\begin{aligned} &< \frac{1}{63} + \frac{1}{b_N^2} - \frac{1}{b_1^2} + 2\sum_{n=2}^{N} \frac{1}{b_n} + \left(1 - \frac{1}{b_N^2} - \frac{1}{b_N}\right) \\ &< 1 + \frac{1}{63} - \frac{1}{4} + 2\left[R^{\infty}_{\mathit{NFD}} - 1 - \tfrac{1}{2}\right] \;=\; 2R^{\infty}_{\mathit{NFD}} - \tfrac{563}{252} \;\approx\; 1.147928... \end{aligned} \tag{7.23}$$

For $\alpha \geqslant 2$, a bound for $c(\alpha)$ follows along similar lines (use lemma B13 to derive the last inequality).

$$\begin{aligned} c(\alpha \geqslant 2) \;&<\; \frac{2}{b_2} + \sum_{n=1}^{N-1} \left(\frac{1}{b_n + 1}\right)^2 + (1 - W_{\text{singleton}}) \\ &< \frac{2}{b_2} + \frac{1}{b_N^2} - \frac{1}{b_1^2} + 2\sum_{n=2}^{N} \frac{1}{b_n} + \left(1 - \frac{1}{b_N^2} - \frac{1}{b_N}\right) \\ &< 1 + \frac{2}{b_2} - \frac{1}{b_1^2} + 2\left[\frac{1}{b_2} + \frac{1}{b_2^2}\right] \;<\; 1 + \frac{3}{b_2} \end{aligned} \tag{7.24}$$

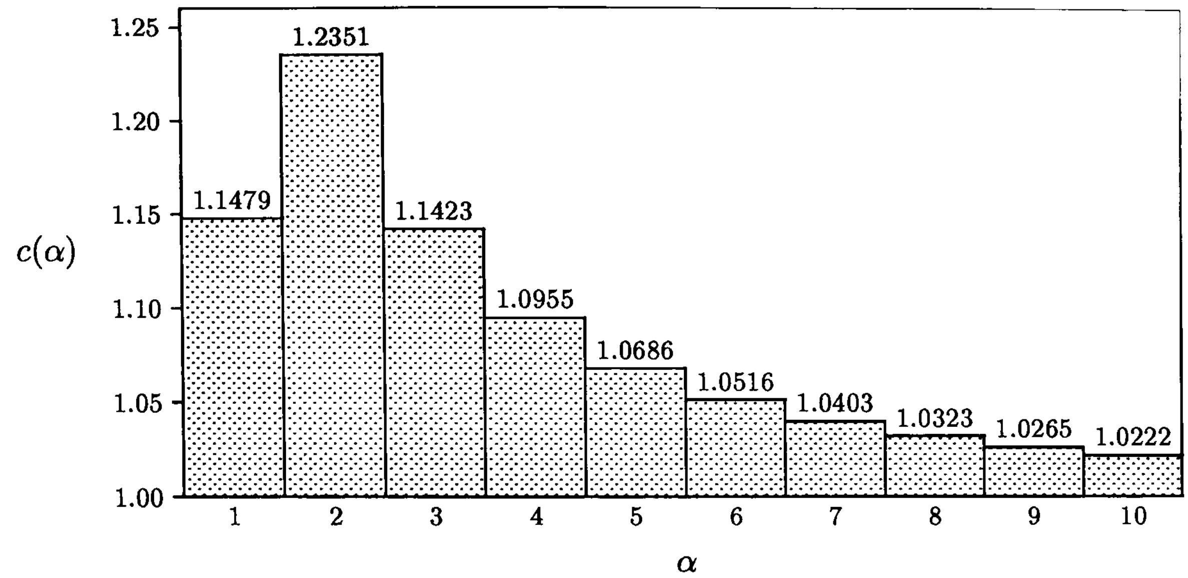


Diagram 7.5. Upper bound for NFD-constant for $\mathcal{L} \subset \langle 0, \frac{1}{\alpha}]$.

7.9 Comments

1) There are lists such that $FF(\mathcal{L}) < NFD(\mathcal{L})$.

   An example for such a list can be found in diagram 7.1. Packing this list by the first-fit heuristic will produce an optimal packing. By taking multiple copies of this list, the difference can be made as large as desired.

2) There are lists such that $NFD(\mathcal{L}) < FF(\mathcal{L})$.

   We know that there are lists such that $FF(\mathcal{L}) = 17k$ and $OPT(\mathcal{L}) = 10k + 1$. (see comment 3 in section 6.8) Using bound (7.2) gives $NFD(\mathcal{L}) < c(1) + r(1)OPT(\mathcal{L}) < \frac{7}{6} + \frac{61}{36}OPT(\mathcal{L})$, substituting the value for $OPT(\mathcal{L})$ yields $NFD(\mathcal{L}) < \frac{103}{36} + (17 - \frac{1}{18})k$. So that for $k \geqslant 52$ we have $NFD(\mathcal{L}) < 17k = FF(\mathcal{L})$. Again the difference can be made as large as desired.

3) One could derive bounds of the form $NFD < c + \bar{r}(\alpha)OPT$, where $\bar{r}(\alpha)$ is an upper bound for $R^{\infty}_{NFD}(1/\alpha)$, as given in (7.16). For instance $\bar{r}(\alpha) = 1 + \frac{\alpha+2}{(\alpha+1)^2}$. This would make sense since for practical purposes we have to use an upper bound for the ratio.

4) The weighting function used by Baker and Coffman[4] is

$$W_2(x) = \begin{cases} \frac{1}{b_n}, & x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}] \\ \frac{\beta+1}{\beta}x, & x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}] \text{ and } \beta \neq b_n \end{cases} \tag{7.25}$$

   It is easily checked that this function is recurrent. If we denote by $W_1(x)$ the weighting function as defined in (7.7) then $W_1(x) \geqslant W_2(x)$. This is obvious for $x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$ since $\rho_n \geqslant 1$. Further if $x \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}] \subset \langle \frac{1}{b_{n+1}}, \frac{1}{b_n+1}]$ then $b_n + 1 \leqslant \alpha$ and $W_1(x) = \rho_n x = \frac{b_n+2}{b_n+1}x \geqslant \frac{\alpha+1}{\alpha}x = W_2(x)$.

5) As a refinement to lemma 7.7 we note that one can prove that a list that maximises $c(\mathcal{L}) = NFD - \sum f_i/i$ is the list shown in diagram 7.3. Moreover, if $\mathcal{L} \in \langle \frac{1}{\beta+1}, \frac{1}{\alpha}]$ one can prove that

$$c(\mathcal{L}) = NFD - \sum f_i/i \leqslant 1 - \tfrac{1}{\alpha} + \sum_{i=\alpha+1}^{\beta} 1/i \tag{7.26}$$

and diagram 7.3 shows that this bound is tight.

6) The lists in (7.17) also provide instances that establish lower bounds for [the ratio of] FF. Take a list $(k\mathcal{L})$, where $b_{N-1}|k$ and sort it into increasing item size. FF will pack this list into the same number of bins as NFD.

7) The last invariant implies that all bins (except the last, singleton bin) are FF-recurrent. This means that the general bound for *FF* also applies to *NFD*:

$$W \text{ is FF-recurrent } \Rightarrow NFD < 1 + \sum_i W(x_i) \tag{7.27}$$

and with it the bounds (6.3) and (6.5). It further implies that the asymptotic ratio of NFD is not larger than that of FF.

8) With regard to the asymptotic ratio we note that the exact sorting sequence a next-fit algorithm uses is unimportant, as long as the items are grouped such that all items in the same i-interval are consecutive in the [sorted] list. For instance, NFI has the same asymptotic ratio as NFD.

9) Note that the instances that show that the asymptotic ratios are achievable have $CSP_R(\mathcal{L}) = Mat(\mathcal{L})$ and are based upon just *one* optimal pattern.

Chapter 8

First-Fit Decreasing Heuristic

8.1 Introduction

The *First-Fit Decreasing (FFD)* heuristic takes a list $\mathcal{L}$, sorts it into non-decreasing order of item size and places each item, in succession, into the first bin it fits. When an item cannot be placed, a new bin is opened in which this item is placed. As an illustration we give the following example.

List $\mathcal{L} = \{ \underbrace{15,\ldots,15}_{3}, \underbrace{9,\ldots,9}_{3}, \underbrace{8,\ldots,8}_{6}, \underbrace{6,\ldots,6}_{9} \}$ on bins of size 29.

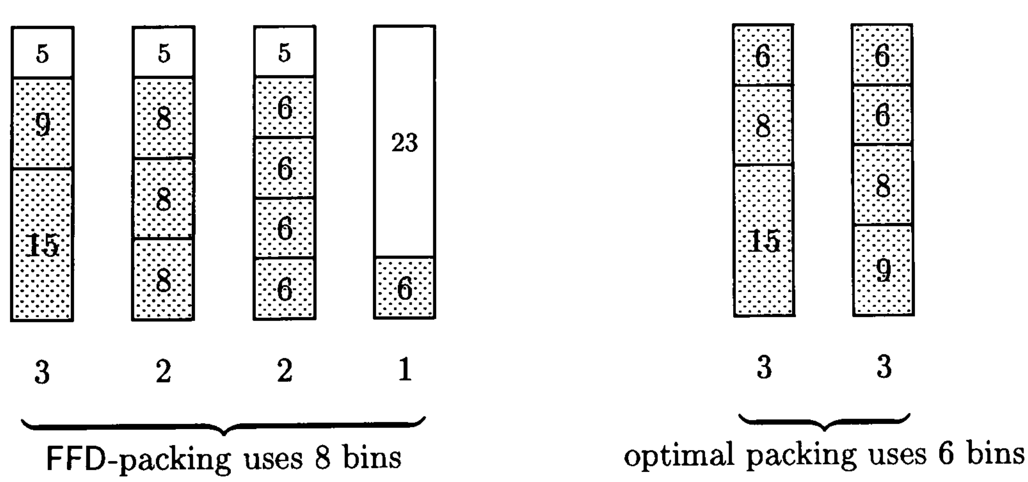


Diagram 8.1. An example of a packing by first-fit decreasing.

The motivation for this algorithm was given by the fact that the FF-algorithm performs worst when the items in the list are arranged in increasing order of item size. To counter this worst-case behaviour, the list is sorted into decreasing sequence, before processing.

The FFD-heuristic was studied by Johnson.[38] He proved that the asymptotic ratio of FFD is 11/9 as opposed to 1.7 for FF. Unfortunately, the theoretical analysis of FFD proved to be both complicated and long-winded. The original proof, in Johnson's thesis, took over 100 pages. In essence, the proof is based on a weighting-function approach as outlined in chapter 5, and an extensive case-analysis. However, this approach fails in a couple of cases and a complicated set-theoretical approach was used to fill in these gaps. As a consequence of the length of the proof, only an outline was published in the normal literature.[37]

It was not until 1985 that a shorter proof was published by Baker.[3] This proof, although shorter (22 pages), relies on much the same techniques and is still rather complicated.

Finally, in 1991 a proof, condensed into 11 pages, was published by Yue.[72] In this paper more refined weighting functions were used covering most cases and some set-theoretical methods to cover the few remaining ones. A comment is made in this paper expressing the belief that this probably is the final sharpening of the worst-case bound for FFD.

Every proof improved upon the constant, as illustrated by table 8.1.

Year	Author	Bound
1973	D. S. Johnson	$FFD \leqslant 4 + \frac{11}{9} OPT$
1985	B. S. Baker	$FFD \leqslant 3 + \frac{11}{9} OPT$
1991	M. Yue	$FFD \leqslant 1 + \frac{11}{9} OPT$

Table 8.1. FFD-bounds in literature.

In this chapter we will study the worst-case performance of the FFD-heuristic in relation to the solution value of the LP-relaxation of the associated cutting stock problem. We will show that one can improve the worst-case bound to [1]

$$FFD < 1 + \tfrac{11}{9} CSP_R, \tag{8.1}$$

which implies

$$FFD \leqslant \tfrac{8}{9} + \tfrac{11}{9} OPT. \tag{8.2}$$

Examples to show that the FFD-bound is asymptotically tight can be found in the literature. Johnson[37] gives an example that achieves $FFD = \frac{11}{9} OPT$, which is replicated in diagram 8.2.

[1]This bound is the best possible (in terms of CSP_R) by lemma 5.12.

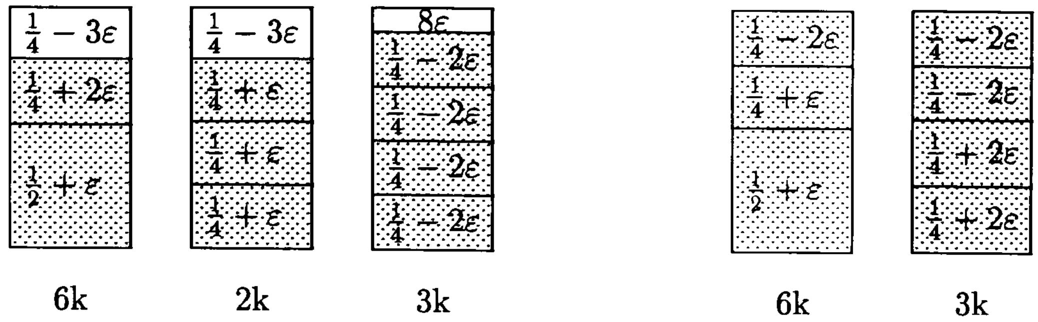


Diagram 8.2. Johnson's example for $FFD{=}\frac{11}{9}OPT$ $(0 < \varepsilon < \frac{1}{40})$. [2]

Yue[72] cites (he does not give) an example to show that the constant in bound (8.2) must be at least $\frac{5}{9}$. An example for this would be the list $\{3,3,2,2,2,2\}$ on bins of size 7, which FFD packs into 3 bins and can be packed optimally in 2 bins.

In the process of proving (8.1) new worst-case examples were found. The list in diagram 8.1 shows that the constant in (8.2) must be at least $\frac{6}{9}$. An example to show that $FFD = \frac{6}{9} + \frac{11}{9}OPT$ is achievable for [arbitrary] large values of OPT is given in diagram 8.26 (page 113). An example to show that $FFD = \frac{22}{27} + \frac{11}{9}CSP_R$ is achievable for [arbitrary] large values of CSP_R is given in diagram D.10 (page 202).

A drawback of the proofs that are published in the literature, is that they offer little or no insight into the structure of lists which exhibit such worst-case behaviour. This would seem to be a prerequisite if one intended to refine and improve upon the heuristic.

The major difficulty encountered in the analysis of the FFD-heuristic is the presence of items with a size in $\langle 1/2, 1]$. As soon as we restrict ourselves to lists with item sizes in $\langle 0, 1/2]$ the analysis simplifies considerably.

In our analysis of the FFD-heuristic we will use the ideas and approach as outlined in chapter 5. These focus on identifying the structure of recurrent patterns and choosing a weighting function accordingly. Furthermore, we take the value of the LP-relaxation as a reference point, rather than the optimal (integer) solution-value. That is, we aim to derive bounds of the form

$$FFD \leqslant c + r\,CSP_R \tag{8.3}$$

Before we start the actual analysis of FFD we investigate certain reductions that can be made a priori. This leads to a simple algorithm to preprocess the list. It further shows, that when the

[2] Johnson[37] states that ε must satisfy $0 < \varepsilon < \frac{1}{12}$. This is incorrect. For FFD to pack exactly 4 items of size $\frac{1}{4} - 2\varepsilon$ in one bin, as is shown in diagram 8.2, we require that $\frac{1}{5} < \frac{1}{4} - 2\varepsilon$, so that ε must satisfy $\varepsilon < \frac{1}{40}$.
Note that the example implies that for every list with smallest item x, with $1/5 < x < 1/4$ there is a configuration that achieves the ratio of $11/9$.

smallest item in the list $x \geqslant \frac{1}{3}$, that FFD is optimal, i.e. $FFD = CSP_I$. As is illustrated in chapter 9, this preprocessing algorithm is a useful addition to the solution process of a cutting stock problem.

After this preprocessing we examine the structure of the bin configuration into which a list of items will pack. Given the number of bins a list packs into we can modify the list to give a new list which packs into the same number of bins but whose LP-solution value is less than (or equal to) that of the original list. We can think of this as identifying lists that maximise $c(\mathcal{L}) = FFD(\mathcal{L}) - CSP_R(\mathcal{L})$. It is interesting to note that the lists that have a 'minimal' LP-solution value represent instances which normally would be classified as cutting stock problems, rather than bin-packing problems.

A simple upper bound will show that we may restrict ourselves to lists with smallest item $x > \frac{2}{11}$. The interval $\langle \frac{2}{11}, \frac{1}{3} \rangle$ is broken down to give three principal cases.

$x \in \langle \frac{1}{4}, \frac{1}{3} \rangle$ This case is the easiest to deal with. The results follow almost directly once the structure of the minimal list has been identified.

$x \in \langle \frac{1}{5}, \frac{1}{4}]$ For this case we first investigate the structure of bins with items in $\langle \frac{1}{2}, 1]$ in a minimal configuration. It turns out that these bins contain only two items and we investigate this case by conditioning on the size of the smallest item in these bins. This gives a total of 6 subcases.

$x \in \langle \frac{2}{11}, \frac{1}{5}]$ For this case we do a similar analysis of the minimal structure. We first condition on the size of the largest item, which gives two subcases. Each of these cases is further analysed by conditioning on the size of the 2$^{\text{nd}}$ item placed in the first bin. This gives a total of 11 subcases.

The above division of $\langle \frac{2}{11}, \frac{1}{3} \rangle$ is a division which can also be found in Baker[3] and in Yue.[72] Each subcase gives rise to a 'minimal' configuration, for which a bound of the form (8.3) is derived.

Structure of the chapter

The structure of this chapter is as follows. In section 8.2 we investigate a preprocessing algorithm, which shows that we can restrict ourselves to lists with sizes in $[x, 1 - 2x]$, where $x < \frac{1}{3}$ is the size of the smallest item. In section 8.3 we examine the bin configuration into which a 'minimal' list will pack. This leads to some assumptions which will be used in the subsequent sections. A naïve upper bound, based on the amount of material to be packed, is derived in section 8.4. This proves (8.1) for $x \leqslant \frac{2}{11}$. It also gives lower bounds on the value of CSP_R for the subsequent cases.

In sections 8.5 and 8.6 we analyse the structure of recurrent patterns and the weighting function that follows from it. In the subsequent sections, 8.7–8.9, we analyse the configuration for the three principal cases as listed previously.

Finally, in section 8.10 we derive parametric worst-case bounds for lists with sizes in the interval $\langle 0, 1/\alpha]$.

8.2 FFD-preprocessing

Under certain conditions FFD produces a packing in which the first bin(s) packed are optimal. In the analysis we will use the equivalence between a bin-packing problem and a cutting stock problem:

$$BPP(\mathcal{L}) \;\cong\; CSP(m, L, \mathbf{d}, \mathbf{f}) \tag{8.4}$$

By appendix A we know that there exists a list of integers that will define exactly the same packing problem. If the original list contains m different item sizes, then the list, scaled to L, will contain f_i items of size d_i on which we can define a cutting stock problem (which is equivalent to the original bin-packing problem).

$$\{x_1 \geqslant \cdots \geqslant x_n\} \text{ on bins of size } 1 \;\cong\; \{\, \underbrace{d_1, \ldots, d_1}_{f_1 \text{ copies}} > \cdots > \underbrace{d_m, \ldots, d_m}_{f_m \text{ copies}} \,\} \text{ on bins of size } L \tag{8.5}$$

The preprocessing algorithm can be thought of as an operator that tries to increase the quantity $c(\mathcal{L}) = FFD(\mathcal{L}) - CSP_R(\mathcal{L})$. It does so by identifying bins and patterns common to FFD, CSP_I and CSP_R.

We first give two lemmas relating the IP-solution to the FFD-solution. We then generalise these for the LP-solution. Denote by $\mathbf{a}_1^{\mathsf{FFD}}$ the pattern corresponding to the first bin packed by FFD.

8.1 Lemma $d_1 + d_m > L \Rightarrow CSP_I(\mathbf{f}) = 1 + CSP_I(\mathbf{f} - \mathbf{a}_1^{\mathsf{FFD}})$

Proof. *The condition implies that any bin containing an item d_1 cannot have another item in it. This is the case for any packing, in particular an optimal packing. Since d_1 is the largest item it will end up in the first bin packed by FFD. Therefore this bin is common to both FFD and the optimal integer solution, and the lemma follows.* □

8.2 Lemma $d_1 + 2d_m > L \Rightarrow CSP_I(\mathbf{f}) = 1 + CSP_I(\mathbf{f} - \mathbf{a}_1^{\mathsf{FFD}})$

Proof. *By lemma 8.1, we can assume $L - d_m < d_1 + d_m \leqslant L$. So exactly one other item will fit in the bin with item d_1. After packing d_1, FFD will pack an item of size $z = \max\,\{d_i \mid d_1 + d_i \leqslant L\}$*

in the first bin.

$$\begin{bmatrix} y \\ d_1 \end{bmatrix} \leftrightarrow \begin{bmatrix} z \\ \cdot \end{bmatrix} \quad \text{gives} \quad \begin{bmatrix} z \\ d_1 \end{bmatrix} \text{ and } \begin{bmatrix} y \\ \cdot \end{bmatrix} \tag{8.6}$$

Suppose that in CSP_I we have an item with size $0 \leqslant y < z$ placed in the bin with item d_1, as shown in (8.6). We can swap the items y and z from the bins they have been packed in and still maintain optimality. We may therefore assume that the first FFD-pattern is an optimal pattern for CSP_I and the lemma follows. □

Note that the condition $d_1 + 2d_m > L$ implies $d_1 > L/3$. We now turn to equivalent lemmas for the LP-solution. For this we will need lemma 2.3 (page 13).

8.3 Lemma $\alpha = \lfloor L/d_1 \rfloor = \lfloor L/d_m \rfloor \Rightarrow$ *FFD* $\leqslant 1 - 1/\alpha + CSP_R \Rightarrow$ *FFD is optimal.*

Proof. *Let $f = \sum f_i$ be the total number of items. It is not difficult to see that FFD $= \lceil f/\alpha \rceil$. A feasible dual multiplier for CSP_R is given by $\mathbf{u} = \mathbf{e}/\alpha$, so that $CSP_R \geqslant f/\alpha$. A feasible solution to the primal is given by the patterns $\alpha \mathbf{e}_i$, used f_i/α times. This gives an upper bound for the optimal solution value of f/α. Ergo $CSP_R = f/\alpha$. The expression FFD $-$ CSP_R is maximised for $f \equiv 1 \pmod{\alpha}$ with value $1 - 1/\alpha$. This proves the first part of the lemma. The second part follows from the elementary bound.* □

For the following lemmas, as a consequence of lemma 8.3, we may assume that $m \geqslant 2$.

8.4 Lemma $d_1 + d_m > L \Rightarrow CSP_R(\mathbf{f}) = 1 + CSP_R(\mathbf{f} - \mathbf{a}_1^{\text{FFD}})$

Proof. *The only pattern that can satisfy the demand for items d_1 is pattern $\mathbf{e}_1$ and must therefore be used at least once. Since $\mathbf{e}_1 = \mathbf{a}_1^{\text{FFD}}$ the lemma follows by lemma 2.3.* □

8.5 Lemma $d_1 + 2d_m > L$ and $d_1 > L/2 \Rightarrow CSP_R(\mathbf{f}) = 1 + CSP_R(\mathbf{f} - \mathbf{a}_1^{\text{FFD}})$

Proof. *By lemma 8.4 we can assume $L - d_m < d_1 + d_m \leqslant L$, so that any pattern that can satisfy the demand for items d_1 can contain at most two items. As before denote by z the item that FFD packs in the 1st bin, where the largest item d_1 is placed. We will show that there is an optimal LP-solution in which the pattern corresponding to $[d_1, z]$ is used at least once. To do this we will make use of the following three clauses to distinguish between cases.*

c1. *There is a pattern active with a z-item and no d_1-item.*

c2. *There is a pattern active with a d_1-item as the only item.*

c3. *There is a pattern active with a d_1-item and an item $y < z$.*

We now investigate the various combinations of these clauses.

A) If c1 and c2 are both true then we can construct a solution with the same solution-value, but with pattern $[d_1, z]$ active by the following conversion, where $\lambda = \min(x_1, x_2) > 0$.

$$\begin{matrix} d_1: \\ z: \\ \end{matrix} \begin{bmatrix} 1 \\ 0 \\ \mathbf{0} \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ a \\ \mathbf{a} \end{bmatrix} x_2 = \begin{bmatrix} 1 \\ 1 \\ \mathbf{0} \end{bmatrix} \lambda + \begin{bmatrix} 1 \\ 0 \\ \mathbf{0} \end{bmatrix} (x_1 - \lambda) + \begin{bmatrix} 0 \\ a-1 \\ \mathbf{a} \end{bmatrix} \lambda + \begin{bmatrix} 0 \\ a \\ \mathbf{a} \end{bmatrix} (x_2 - \lambda) \tag{8.7}$$

For convenience we have condensed all the other sizes as **0** *and* **a**, *since this does not affect the rationale. We repeat this conversion until either c1 or c2 is false.*

B) If c1 and c3 are both true then we use the following conversion, where $\lambda = \min(x_1, x_2) > 0$.

$$\begin{matrix} d_1: \\ z: \\ y: \\ \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ a \\ b \\ \mathbf{b} \end{bmatrix} x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ \mathbf{0} \end{bmatrix} \lambda + \begin{bmatrix} 1 \\ 0 \\ 1 \\ \mathbf{0} \end{bmatrix} (x_1 - \lambda) + \begin{bmatrix} 0 \\ a-1 \\ b+1 \\ \mathbf{b} \end{bmatrix} \lambda + \begin{bmatrix} 0 \\ a \\ b \\ \mathbf{b} \end{bmatrix} (x_2 - \lambda) \tag{8.8}$$

We repeat this conversion until either c1 or c3 is false.

After converting the optimal LP-solution, using (8.7) and (8.8), we have one of the following.

C) If c2 and c3 are both false then the demand for d_1-items is satisfied by the pattern $[d_1, z]$ only.

D) If either c2 or c3 is true then c1 must be false. This means that every pattern with a z-item contains a d_1-item, and thus that the demand for z-items is satisfied by the pattern $[d_1, z]$ only.

In both cases we have that the pattern $[d_1, z]$ is used at least once in the optimal LP-solution and we can apply lemma 2.3 to prove the lemma. □

In lemma 8.5 we conditioned on d_1 being a 1-item, in order not to have to consider the pattern $[d_1, d_1]$. The case when $d_1 \leqslant L/2$ will now be dealt with by the following lemma.

8.6 Lemma $d_1 + 2d_m > L$ and $d_1 \leqslant L/2 \implies$ *FFD* $\leqslant 5/6 +$ *CSP*$_R$ $\implies$ *FFD is optimal.*

Proof. *First note that the conditions imply $d_1 > L/3$ and $d_m > L/4$, so that all items are either 2-items or 3-items. In particular the largest item is a 2-item and the smallest item is a 3-item (since $m \geqslant 2$ by lemma 8.3. From $d_1 + 2d_m > L$ we have that any pattern with an item d_1 can contain at most one other item. Assume that there are $m_2 \geqslant 1$ different 2-items and $m_3 \geqslant 1$ different 3-items ($m_2 + m_3 = m$). First we will show that we may assume that the only pattern that is active in the LP-solution and contains an item d_1 is the pattern $[d_1, d_1]$.*

- *pattern $[d_1]$ cannot be active; since we can replace it, and reduce the solution value, as follows.*

$$d_1: \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 0 \end{bmatrix} \times \frac{1}{2} \tag{8.9}$$

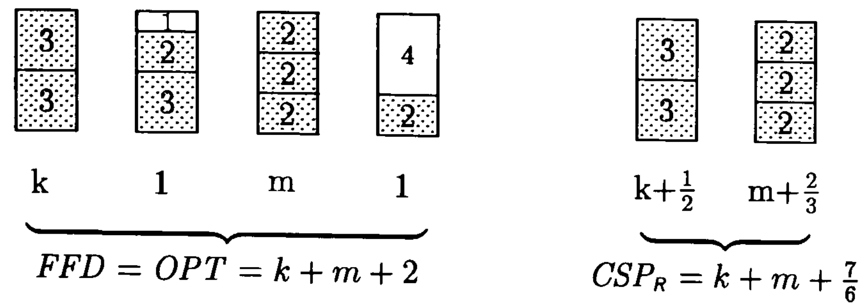


Diagram 8.3. Worst-case instance for $\mathcal{L} \subset [\frac{1}{3}, 1]$; $FFD = \frac{5}{6} + CSP_R$.

- *pattern $[d_1, z]$ with $z < d_1$ can also be replaced, without increasing the solution value.*

$$\begin{matrix} d_1: \\ z < d_1: \\ \end{matrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \times \frac{1}{2} + \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix} \times \frac{1}{2} \tag{8.10}$$

So we may assume that the demand for 2-items is satisfied by the patterns $2\mathbf{e}_i$ only with usage $f_i/2$, $(i = 1, \ldots, m_2)$. In a similar fashion one can show that we may assume that the demand for 3-items is satisfied by the patterns $3\mathbf{e}_i$ with usage $f_i/3$, $(i = m_2 + 1, \ldots, m)$. This means that the LP-solution has value $CSP_R = m_2/2 + m_3/3$. It is not difficult to show that FFD will use the following number of bins;

$$FFD = \begin{cases} \frac{m_2}{2} + \lceil \frac{m_3}{3} \rceil, & m_2 = even \\ \frac{m_2+1}{2} + \lceil \frac{m_3-1}{3} \rceil, & m_2 = odd \end{cases} \tag{8.11}$$

The expression $FFD - CSP_R$ is maximised for $m_2 \equiv 1 \pmod 2$ and $m_3 \equiv 2 \pmod 3$ with value 5/6. This proves the first part of the lemma. The second part follows from the elementary bound. □

An example to show that the bounds in lemma 8.6 are tight is easily constructed and shown in diagram 8.3.

The lemmas can be converted directly into a reduction procedure. This is best described in the form of a flowchart as shown in diagram 8.4. Denote $c(\mathcal{L}) = FFD(\mathcal{L}) - CSP_R(\mathcal{L})$. For a given list $\mathcal{L}$, any of the reductions implied by lemma 8.4 or 8.5 gives a list $\mathcal{L}'$ with $c(\mathcal{L}') = c(\mathcal{L})$. So that after repeatedly applying the reduction procedure we either prove that $FFD(\mathcal{L}) = OPT(\mathcal{L})$ or end up with a list $\mathcal{L}' \subset \mathcal{L}$ such that $c(\mathcal{L}') = c(\mathcal{L})$ and $x'_1 + 2x'_n \leqslant 1$.

8.7 Lemma $x_n \geqslant \frac{1}{3} \Rightarrow FFD \leqslant \frac{5}{6} + CSP_R \Rightarrow FFD = OPT$. [3]

Proof. *For a given list $\mathcal{L}$ with smallest item $x_n \geqslant 1/3$ let $\mathcal{L}'$ be the list after applying the pre-processing algorithm. If $x_n > \frac{1}{3}$ then $\mathcal{L}'$ is empty, FFD is optimal and the maximum value of $c(\mathcal{L})$*

[3] *We note that lemma 8.7 is, in a sense, the best possible; comment 1 (p. 95) implies that for every $x_n < 1/3$ there are instances for which FFD is not optimal.*

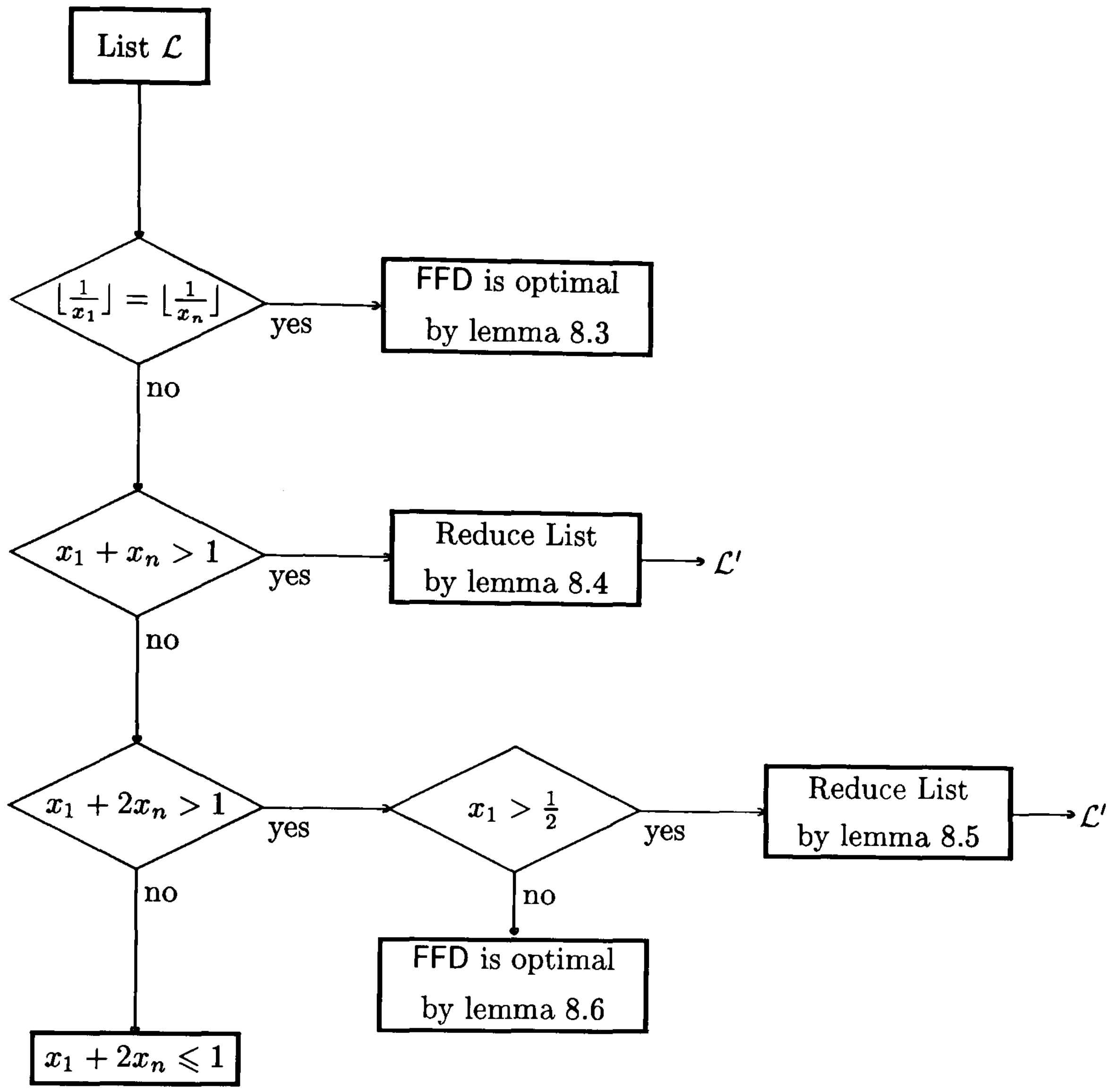


Diagram 8.4. Flowchart for the FFD-preprocessing algorithm.

is given by lemma 8.6 as $5/6$. *If* $\mathcal{L}'$ *contains* $n' \geqslant 1$ *items then* $x_1' = x_n' = \frac{1}{3}$ *must hold. So that* $FFD(\mathcal{L}') = \lceil n'/3 \rceil$, $CSP_R(\mathcal{L}') = n'/3$ *and* $c(\mathcal{L}) = c(\mathcal{L}') \leqslant 2/3$. □

Lemma 8.7 and the preprocessing algorithm [depicted in diagram 8.4] give the following two assumptions that can be made on the lists that need to be investigated in order to prove (8.1).

8.8 Assumption $x_n < \frac{1}{3}$

8.9 Assumption $x_1 + 2x_n \leqslant 1$

In the subsequent sections we determine lists that maximise $FFD(\mathcal{L}) - r\,CSP_R(\mathcal{L})$, for a given value of $r > 1$ (in particular $11/9$). It is easily seen that assumptions 8.8 and 8.9 also hold for such a maximising list.

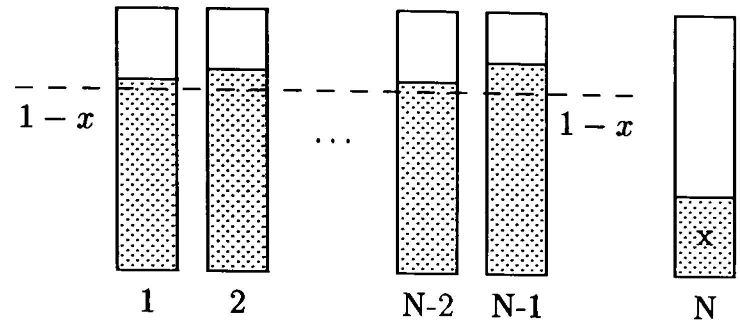


Diagram 8.5. Singleton-bin configuration.

8.3 Minimal Bin-Configuration

For a list that maximises the gap $c(\mathcal{L}) = FFD(\mathcal{L}) - r\,CSP_R(\mathcal{L})$, $(r \geqslant 1)$ we can make the following assumptions about the bin configuration it packs into.

8.10 Assumption *The smallest item in the list is packed into the last bin as a singleton item.*

Proof. *Suppose a FFD-packing of $\mathcal{L}$ uses N bins, and denote by x the first item placed in the last bin. Create a list $\mathcal{L}'$ (with smallest item x) by deleting all items from $\mathcal{L}$ which are packed after item x. Since $FFD(\mathcal{L}') = FFD(\mathcal{L})$ and $CSP_R(\mathcal{L}') \leqslant CSP_R(\mathcal{L})$ we have a list with a gap at least as large. By assumption item x in the list $\mathcal{L}'$ is packed in the last bin as a singleton item.* □

Since we are looking for a list that maximises the gap, we may assume that the list, when packed by FFD, has a structure as depicted in diagram 8.5. We will refer to the item x as the *critical item.* The last bin will be referred to as a *singleton bin.* Note that *all* bins, except the singleton bin, are filled to a level strictly larger than $1 - x$, by virtue of the FFD-algorithm.

Now assume that $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$, where $\beta \in \mathbb{N}^+$. We can then make the following assumptions on the minimal configuration.

8.11 Assumption *Any bin with β items is a bin with β items of size x.*

Proof. *Assume that a bin contains β items. Now replace each item by an item of size x. This new list will pack in the same number of bins as is illustrated in diagram 8.6. Since each of the items that we replaced had size $\geqslant x$, we have created a new list with a smaller (or equal) LP-solution value. We shall refer to this as the* cutting principle, *since we 'cut' sizes down to size x.* □

8.12 Assumption *There is no bin with largest item y such that $x < y \leqslant \frac{1-x}{\beta-1}$.*

Proof. *If there is a bin with largest item $y \leqslant \frac{1-x}{\beta-1}$ then this bin must contain exactly β items, which contradicts assumption 8.11* □

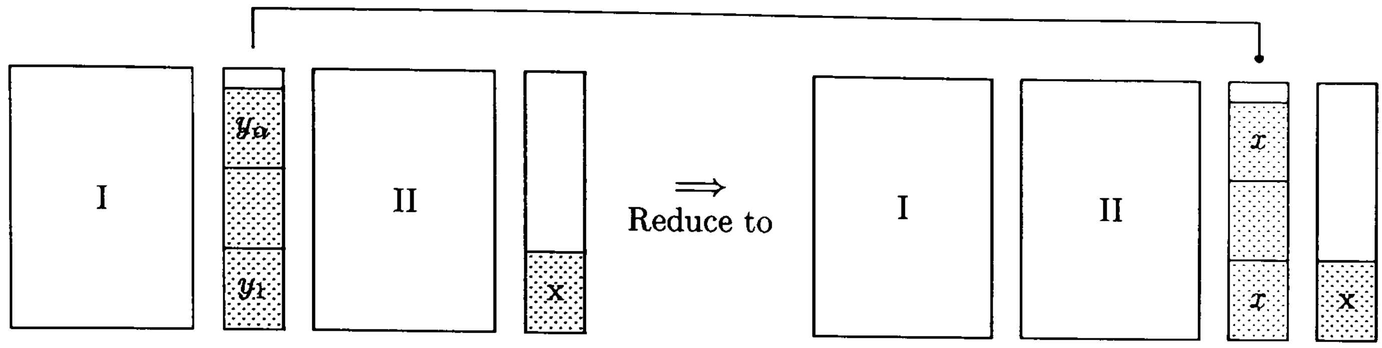


Diagram 8.6. Cutting principle

8.13 Assumption *If a bin contains items $y_1, \dots, y_m$ with $m \leqslant \beta - 1$, then $\sum_{i=1}^{m} \lfloor y_i/x \rfloor \leqslant \beta - 1$.*

Proof. *'Cut' each item y_i into $\lfloor y_i/x \rfloor$ items of size x giving β items x, and use assumption 8.11* □

8.14 Assumption *If a bin contains i items with size in $\langle \frac{1-x}{i}, \frac{1}{i}]$ then these items have the same size.*

Proof. *If y is the smallest item in the bin, replace all i items by an item of size y. All lower indexed bins are filled to a level $> 1-y$, so that FFD will not place any of these items in a previous bin. Furthermore, the bin is still filled to a level $> 1-x$, so that FFD will not place any additional items in this bin. Therefore FFD will pack the new list in the same number of bins.* □

8.15 Assumption *If two consecutive bins have a largest [and first] item with size in $\langle \frac{1-x}{i}, \frac{1}{i}]$ then all the items in the first bin and the first item in the second bin have the same size.*

Proof. *If y is the first item placed in the second bin, then we can replace all items in the first bin by items of size y. By the rationale of assumption 8.14, FFD will pack the new list in the same number of bins.* □

8.16 Assumption *If there is a bin with largest item $y \in \langle \frac{1-x}{i}, \frac{1}{i}]$ then there are no bins with largest item in $\langle \frac{1}{i}, \frac{1-y}{i-1}]$.*

Proof. *By contradiction, assume that there is a bin with largest item in $\langle \frac{1}{i}, \frac{1-y}{i-1}]$. After FFD has placed the first $i-1$ items, the bin is filled to a level $\leqslant (i-1) \times \frac{1-y}{i-1} = 1-y$, so that FFD will place another item with size at least y in this bin. We can now replace this bin by a bin with i items of size y. This does not affect the number of bins that FFD uses. The assumption is therefore false and the lemma follows.* □

Note that all of the reductions, associated with the assumptions, create a new list which FFD packs into the same number of bins and has a CSP_R-value which is not larger. Therefore all these reductions do not decrease and may increase the gap $c(\mathcal{L}) = FFD - rCSP_R$.

8.4 Naïve upper bound

We can use the singleton-bin configuration (diagram 8.5) to derive an upper bound for *FFD*, by means of a simple area-argument. Denote by l_j the level to which bin j is filled, by N the number of bins FFD uses, and by *Mat* the total amount of *material* to be packed, i.e. $Mat = \sum x_i$. Then $l_j > 1 - x$ for $j = 1, \ldots, N-1$ and $l_N = x$. This gives

$$FFD \geqslant 2 \;\Rightarrow\; (N-1) \times (1-x) + x < \sum_{j=1}^{N} l_j = \sum_{i=1}^{n} x_i = Mat \tag{8.12}$$

and thus the *naïve* bound

$$Mat > x \;\Rightarrow\; FFD < 2 + \frac{1}{1-x}\,(Mat - 1)\,. \tag{8.13}$$

This provides us with (cruder) upper bounds in terms of CSP_R and OPT. In particular

$$Mat > \tfrac{1}{\beta} \geqslant x \;\;\Rightarrow\;\; FFD < 2 + \frac{\beta}{\beta - 1}\,(Mat - 1)\,, \tag{8.14}$$

$$CSP_R > x \;\;\Rightarrow\;\; FFD \leqslant 1 + \left\lceil \tfrac{CSP_R - 1}{1-x} \right\rceil, \tag{8.15}$$

and the unconditional bound

$$FFD \leqslant 1 + \left\lceil \tfrac{OPT - 1}{1-x} \right\rceil. \tag{8.16}$$

Bound (8.13) allows the following corollaries.

8.17 Corollary $x \leqslant \frac{2}{11} < Mat \;\Rightarrow\; FFD < \frac{7}{9} + \frac{11}{9}Mat \;\Rightarrow\; FFD < \frac{7}{9} + \frac{11}{9}CSP_R$.

8.18 Corollary $x \leqslant \frac{2}{11} \;\Rightarrow\; FFD \leqslant \frac{6}{9} + \frac{11}{9}OPT$. [4]

In order to prove (8.1) we will need lower bounds on the value of CSP_R. These can be established using bound (8.15) as follows. For $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$ determine the maximum value of CSP_R such that $\lceil \frac{CSP_R - 1}{1-1/\beta} \rceil < \frac{11}{9}CSP_R$ holds. This gives the bounds in the following corollaries.

8.19 Corollary $x \in \langle \frac{1}{5}, \frac{1}{4}]$ *and* $CSP_R \leqslant 3\frac{1}{4} \;\Rightarrow\; FFD < \frac{17}{18} + \frac{11}{9}CSP_R$

8.20 Corollary $x \in \langle \frac{2}{11}, \frac{1}{5}]$ *and* $CSP_R \leqslant 8\frac{1}{5} \;\Rightarrow\; FFD < \frac{43}{45} + \frac{11}{9}CSP_R$

Once we have proven (8.1) it follows that $c = FFD - \frac{11}{9}OPT$ has a value of $8/9$ or less. Since we are dealing with integers we can make a more definite statement about the nature of *FFD* and

c	FFD	OPT	comments
0/9	$0+11k$	$0+9k$	
1/9	$5+11k$	$4+9k$	
2/9	$10+11k$	$8+9k$	
3/9	$4+11k$	$3+9k$	
4/9	$9+11k$	$7+9k$	
5/9	$3+11k$	$2+9k$	
6/9	$8+11k$	$6+9k$	
7/9	$13+11k$	$10+9k$	$x>2/11$
8/9	$7+11k$	$5+9k$	$x>\frac{1+2k}{5+11k}$

Table 8.2. Structure for $FFD = c + \frac{11}{9}OPT$, $k \in \mathbb{N}$.

OPT, depending on the value of c. Using congruences we can derive table 8.2. Using bound (8.16) we can eliminate the case $(FFD, OPT) = (2,1)$ and add the restriction $x > 2/11$, for $c = 7/9$. Similarly we can show that for $c = 8/9$ we have $x > \frac{1+2k}{5+11k}$, which means that $k \geqslant 1$ must hold for $x \leqslant 1/5$.

A direct consequence of corollary 8.17 is that, in order to prove (8.1), we may restrict ourselves to lists with a critical item size $x > \frac{2}{11}$, which gives the following assumption.

8.21 Assumption $x > \frac{2}{11}$

It turns out that it is slightly more convenient in the subsequent analysis to work with open intervals for x of the form $\langle\frac{1}{\beta+1}, \frac{1}{\beta}\rangle$. The reason for this assumption is that when we partition the interval $[x, 1-2x]$ we avoid certain intervals to be empty (see the comments at the beginning of sections 8.7, 8.8 and 8.9). This does not constitute a restriction as is shown by the following assumption.

8.22 Assumption $x \neq \frac{1}{\beta}$

Proof. *Assume the bound $FFD \leqslant c + r\,CSP_R$ holds for all lists with critical item size $x \in \langle\frac{1}{\beta+1}, \frac{1}{\beta}\rangle$. Now suppose that there is a list with critical item size $x = \frac{1}{\beta}$, such that this bound does not hold. Reduce the size of (only) the critical item to $\frac{1}{\beta} - \varepsilon$. For ε sufficiently small. this will give the same values for FFD and CSP_R, and thus produce an instance with critical item $x \in \langle\frac{1}{\beta+1}, \frac{1}{\beta}\rangle$ for which*

[4] *As an addendum to corollary 8.18, note that for $x \in \langle\frac{5}{29}, \frac{2}{11}]$ there are lists which achieve $FFD = \frac{6}{9} + \frac{11}{9}OPT$ (reduce the size of the critical item in diagram 8.1 accordingly).*

the bound does not hold. Contradiction. Ergo, if we have proven a linear bound for all lists with critical item $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta} \rangle$, it is also a bound for $x = \frac{1}{\beta}$. □

8.5 Recurrent Patterns

To apply the analysis of chapter 5, in order to determine a weighting function, we need to determine what the recurrent bins and patterns are in a packing by FFD. To this end we define the following.

8.23 Definition ***(Self-complete bin/pattern.)*** *A bin or pattern is said to be self-complete if it contains items from a set of m different sizes $s_1 > \cdots > s_m$ in quantities $a_1, \ldots, a_m$ satisfying*

$$a_1 = \lfloor 1/s_1 \rfloor \quad \text{and} \quad a_i = \left\lfloor (1 - \sum_{k=1}^{i-1} a_k s_k)/s_i \right\rfloor, \ i = 2, \ldots, m \tag{8.17}$$

Basically each size is duplicated as many times as possible, starting with the largest size.

8.24 Lemma *FFD-recurrent bins are self-complete bins.*

Proof. *Suppose we have two identical bins in a packing by FFD, and let s be the size of the first item placed in these bins. This means that FFD will have placed the maximum number of items of size s in the first bin, before placing the first one of size s in the second, so that the first bin (and thus the second)* must *contain $\lfloor 1/s \rfloor$ items of size s. Repeating this argument for the next size in the bin, we see that this defines a self-complete pattern.* □

8.6 Generic Weighting Function

From chapter 5 we know that we need to choose a weighting function W, such that the weight of any recurrent bin is at least one. Suppose that we have a recurrent pattern/bin which contains i items of the same size $s \in \langle \frac{1}{i+1}, \frac{1}{i}]$. The residual length in the bin is $1 - (i \times s) < 1 - (i \times \frac{1}{i+1}) = \frac{1}{i+1}$. If the critical item size $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$ then the maximum number of additional items k, that FFD can place in this bin, will satisfy $k < \frac{1}{i+1}/x < \frac{\beta+1}{i+1}$, so that $k \leqslant \lceil \frac{\beta-i}{i+1} \rceil$ holds (see table 8.3). Now suppose that we have a recurrent bin with largest item $y \in \langle \frac{1}{i+1}, \frac{1}{i}]$. If y is such that we can place no additional items in the bin, then the requirement on the weighting function is $i \times W(y) \geqslant 1$, and if we can place exactly one item the requirement is $i \times W(y) + W(x) \geqslant 1$.

To prove (8.1) we only need to consider the values $\beta \in \{3, 4, 5\}$, by assumptions 8.8 and 8.21. From table 8.3 we can see that, except for the 1-items, every i-complete bin can contain at most

$\beta\backslash i$	1	2	3	4	5	6	7	8	9	10
1	0									
2	1	0								
3	1	1	0							
4	2	1	1	0						
5	2	1	1	1	0					
6	3	2	1	1	1	0				
7	3	2	1	1	1	1	0			
8	4	2	2	1	1	1	1	0		
9	4	3	2	1	1	1	1	1	0	
10	5	3	2	2	1	1	1	1	1	0

Table 8.3. Maximum number of additional items that FFD can place in an i-complete bin for critical item size $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$.

one additional item. This leads, almost naturally, to the following weighting function[5]

$$\beta \in \{3,4,5\} \text{ and } s \in [x, \tfrac{1}{2}] \;\Rightarrow\; W(s) = \begin{cases} \frac{1}{i}, & s \in \langle \frac{1-x}{i}, \frac{1}{i}] \text{ and } 2 \leqslant i \leqslant \beta - 1 \\ \frac{1-W_x}{i}, & s \in \langle \frac{1}{i+1}, \frac{1-x}{i}] \text{ and } 2 \leqslant i \leqslant \beta - 2 \\ W_x, & s \in [x, \frac{1-x}{\beta-1}] \end{cases} \tag{8.18}$$

where W_x is to be chosen such that $\frac{1}{\beta} \leqslant W_x \leqslant \frac{1}{\beta-1}$, in order for $W(s)$ to be non-decreasing. We shall refer to this weighting function as the *generic weighting function* in subsequent sections. Note that the division of the interval $[x, \frac{1}{2}]$ into $[x, \frac{1-x}{\beta-1}], \ldots, \langle \frac{1-x}{2}, \frac{1}{2}]$ is a proper division. We note that the choice of $W_x = \frac{1}{\beta}$ gives a weighting function as introduced and used by Baker.[3]

From the previous we can see why the difficulty in the analysis of FFD lies in the 1-items. It is not obvious how one should choose a weight for these. This depends on how many items are actually placed in a 1-bin. The answer, as we shall see, lies in an even more detailed analysis of the bin configuration, eliminating certain 1-bin configurations.

Comment If the largest item in the list, $\varphi \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ then the condition $\beta \leqslant 2\alpha + 1$ ensures that every i-complete bin contains at most one additional item. If this is the case then we can use (8.18) as a recurrent weighting function for all the items in the list.

[5]This weighting function can be interpreted as a refinement of the weighting function (7.6), which was used for NFD. Using this weighting function; viz. $W(s) = 1/\lfloor 1/s \rfloor = 1/\alpha_i$ one can prove the bound $FFD < 2 + \sum f_i/\alpha_i$, as is done in appendix E.2.

8.7 Case $x \in \langle \frac{1}{4}, \frac{1}{3} \rangle$

We divide the interval $[x, 1-2x]$ into two sub-intervals as follows

$$[x, 1-2x] : \begin{cases} I_1 = \left\langle \frac{1-x}{2}, 1-2x \right], & \text{2-items} \\ I_2 = \left[x, \frac{1-x}{2} \right], & \text{2 and 3-items} \end{cases} \tag{8.19}$$

Note that the list contains no 1-items, since $1 - 2x < \frac{1}{2}$. Note further that the condition $x < \frac{1}{3}$ is necessary to ensure that the interval I_1 is non-void. Any list with critical item $x \in \langle \frac{1}{4}, \frac{1}{3} \rangle$ packs into the following configuration.

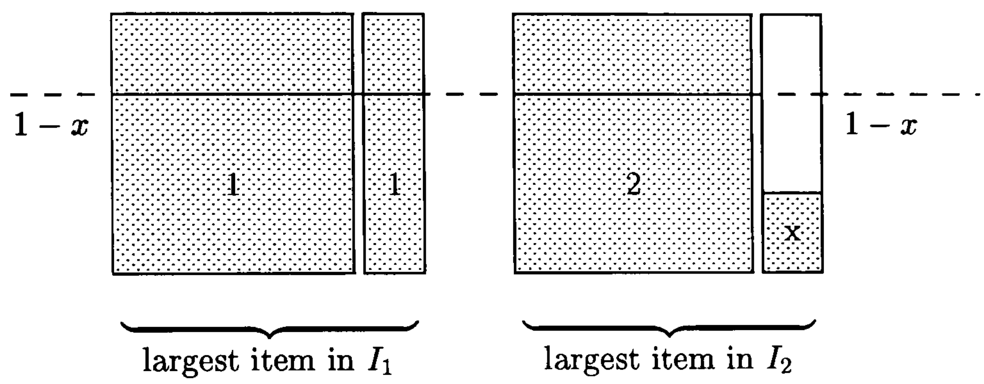


Diagram 8.7. General bin-configuration for critical item $x \in \langle \frac{1}{4}, \frac{1}{3} \rangle$.

8.7.1 Minimal configuration

We apply the assumptions of section 8.3 to reduce the general configuration to a minimal one.

1. Block 1 consists of bins with 2 items.

   There cannot be three items in these bins by assumption 8.11, and there must be at least two since the size of the largest item is strictly less than $\frac{1}{2}$.

2. Block 2 consists of bins with $\beta = 3$ items of size x, by assumption 8.12.

3. All items in the interval I_1 have the same size, by assumptions 8.14 and 8.15.

In addition to this the following assumptions can be made. The list must have at least one item in I_1, otherwise FFD is optimal. The last bin in block 1 contains an item in I_1 and an item in I_2. If this were not the case and both are in I_1 we could reduce the smallest one to a size $\frac{1-x}{2}$, without affecting the number of bins that FFD uses. Combining the assumptions gives the structure, as shown in diagram 8.8, for a minimal configuration.

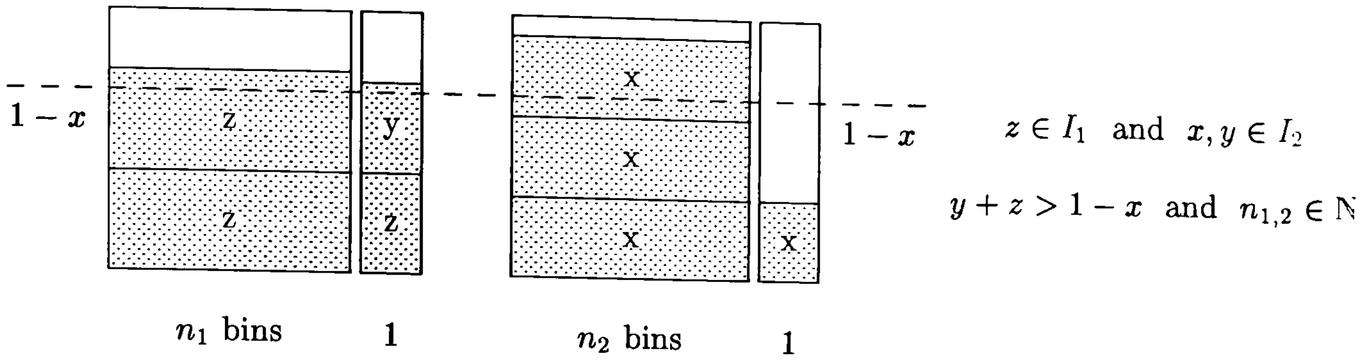


Diagram 8.8. Minimal bin-configuration for $x \in \langle \frac{1}{4}, \frac{1}{3} \rangle$.

8.7.2 Weighting function

To derive a bound we use the following weights for the items.[6]

$$W(z) = \tfrac{1}{2}, \quad W(y) = \tfrac{7}{18} \text{ and } W(x) = \tfrac{1}{3} \tag{8.20}$$

It is easily verified that all recurrent bins, viz. bins $[z, z]$ and $[x, x, x]$ have a bin weight of one. With these weights we get

$$W(\mathcal{L}) = \sum_i W(x_i) = [2n_1 + 1]\, W(z) + W(y) + [3n_2 + 1]\, W(x) = FFD - \tfrac{7}{9} \tag{8.21}$$

8.7.3 Ratio problem

The asymptotic ratio r can now be determined from the ratio problem;

$$r = \boxed{\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i \leqslant 1 \\ & 0 < x_i \leqslant 1 \end{array}} = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{7}{18}a_2 + \frac{1}{3}a_3 \\ \text{st} & z a_1 + y a_2 + x a_3 \leqslant 1 \\ & a_i \in \mathbb{N},\ y > 1 - x - z \\ & z \in \langle \frac{1-x}{2}, 1 - 2x] \text{ and } x \in \langle \frac{1}{4}, \frac{1}{3} \rangle \end{array}} \tag{8.22}$$

Denote by $\mathcal{A}$ the set of all patterns **a** that are feasible patterns for some realisation of z, y and x. The *knapsack* problem (8.22) can be solved by taking the maximum pattern value over all feasible patterns. All the *profits* in the objective function are positive, which implies that the maximum is achieved for an extremal pattern. So we may solve (8.22) by considering only the extremal patterns of $\mathcal{A}$. These are enumerated in table 8.4 and show that r is equal to 7/6, and achieved by patterns 3 and 4.

[6]The weighting function (8.20) was derived by first determining the asymptotic ratio 7/6, using only the weights for the recurrent items $W(z)$ and $W(x)$. Afterwards $W(y)$ is determined as the largest value such that (8.22) still yields the same asymptotic ratio.

pattern nr	1	2	3 *	4 *	5	6	7
a_1	2	1	1	0	0	0	0
a_2	0	1	0	3	2	1	0
a_3	0	0	2	0	1	2	3
18×value	18	16	21	21	20	19	18

Table 8.4. Extremal pattern set $\mathcal{E}$ of program (8.22)

This gives the following bound

$$W(\mathcal{L}) = \sum_j W(\mathbf{a}_j)x_j^* \leqslant r\,CSP_R = \tfrac{7}{6}CSP_R \tag{8.23}$$

where $\mathbf{a}_j$ represents the j^{th} pattern in the CSP-formulation of the BPP, x_j^* the optimum solution value of the j^{th} pattern, $CSP_R = \sum x_j^*$ and r is the maximum weight that any pattern or bin can have under this weighting function.

8.7.4 Bounds

Combining the expressions in $W(\mathcal{L})$ gives the following bound for the number of bins that FFD uses on a [minimal] configuration.

$$x \in \left\langle \tfrac{1}{4}, \tfrac{1}{3} \right\rangle \;\Rightarrow\; FFD \leqslant \tfrac{7}{9} + \tfrac{7}{6}CSP_R \tag{8.24}$$

which implies

$$x \in \left\langle \tfrac{1}{4}, \tfrac{1}{3} \right\rangle \;\Rightarrow\; FFD \leqslant \tfrac{2}{3} + \tfrac{7}{6}OPT \tag{8.25}$$

It is easily verified that these bounds prove (8.1) and (8.2) for the case $x \in \left\langle \tfrac{1}{4}, \tfrac{1}{3} \right\rangle$.

8.7.5 Problem instance

We now want to find lists that show that these bounds are tight. From the minimal configuration (diagram 8.8) we see that such an instance is determined by the frequency of the items $(n_{1,2})$ and their sizes (x, y, z). This translates into two separate problems.

1. To find values for the item frequencies we set up a *balance* equation between the patterns in the FFD-configuration and the LP-patterns that achieve the asymptotic ratio.
2. To find values for the item sizes we solve the *generator* problem for the extremal pattern set $\mathcal{E}$.

These two problems are solved in the next sections.

8.7.6 Balance

The balance for the minimal configuration is given by

$$\begin{array}{c} z: \\ y: \\ x: \end{array} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} n_1 \\ 1 \\ n_2 \\ 1 \end{bmatrix} = \mathbf{f} = \begin{bmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \text{ with } n_{1,2} \in \mathbb{N} \text{ and } x_{1,2} \geqslant 0 \tag{8.26}$$

The LHS of (8.26) corresponds to the FFD-packing of the f_1 items z, f_2 items y and f_3 items x. The RHS corresponds to the LP-solution (restricted to the patterns that achieve the asymptotic ratio). Solving (8.26) gives

$$\begin{cases} 2n_1 + 1 = x_1 \\ 1 = 3x_2 \\ 3n_2 + 1 = 2x_1 \end{cases} \Rightarrow \begin{cases} x_1 = 1 + 2n_1 \\ x_2 = \frac{1}{3} \\ 3n_2 = 1 + 4n_1 \end{cases} \Rightarrow \begin{cases} x_1 = 5 + 6k \\ x_2 = \frac{1}{3} \\ n_1 = 2 + 3k \\ n_2 = 3 + 4k \end{cases} \tag{8.27}$$

Note that the frequencies of usage of the FFD-patterns must be integer, whereas those of the LP-patterns are only required to be non-negative. Taking this into account gives the values as shown. Note that the solution leaves us with one degree of freedom. The above derivation is condensed in table 8.5, which is the format that will be used subsequently.

	FFD-bins					LP-patterns	
z	2	1	0	0	6k+5	**1**	**0**
y	0	1	0	0	1	**0**	**3**
x	0	0	3	1	12k+10	**2**	**0**
	3k+2	1	4k+3	1		6k+5	$\frac{1}{3}$
	7k+7					**$6k + 5\frac{1}{3}$**	

Table 8.5. Balance for $FFD = \frac{7}{9} + \frac{7}{6}CSP_R$.

That the LP-patterns constitute an optimum solution can be seen as follows. The vector $\mathbf{u} = \mathrm{e}/3$ is a valid dual multiplier, that is all patterns in the extremal pattern set $\mathcal{E}$ have $\langle \mathbf{a}, \mathbf{u} \rangle \leqslant 1$, so that $CSP_R \geqslant \langle \mathbf{f}, \mathrm{e}/3 \rangle = 6k + 16/3$, from which optimality follows.

8.7.7 Generator

We can use the procedure as described in appendix B.7 and solve the IP associated with the extremal-pattern set $\mathcal{E}$ listed in table 8.4. This gives a minimal generator (with respect to the scalar) for the pattern set $\mathcal{A}$.

$$\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^3 \mid 7a_1 + 5a_2 + 4a_3 \leqslant 15\} \tag{8.28}$$

It is easily verified that these sets are equivalent, since they have the same extremal patterns. This immediately gives values for (z, y, x) as $\frac{1}{15}(7, 5, 4)$.

8.7.8 Examples

Having solved the *balance* and *generator* problem, an example to show that bound (8.24) is tight can easily be constructed and is shown in diagram 8.9. An example showing that bound (8.25) is tight is given in diagram 8.10. This is derived by the same process as the previous example. The only difference is that we now set up a balance between the FFD-patterns and the IP-solution. This means that the frequency of usage of the LP-patterns are required to be integer.

8.7.9 Comments

The configurations in diagrams 8.9 and 8.10 can be used to infer the following.

1) The bounds (8.24) and (8.25) are tight for *every* $x \in \langle 1/4, 1/3 \rangle$. Simply take the item sizes as $(1 - 2x, 1/3, x)$ and alter the diagrams accordingly.

2) The ratio of 7/6 is achievable for all critical item sizes $1/7 < x < 1/3$; reduce the size of the singleton item in diagram 8.10. Note that, by (8.12) this is the widest range of x for which this ratio is achievable.

3) Taking $k = 0$ in diagram 8.10 gives the list $\{\frac{3}{7}, \frac{3}{7}, \frac{2}{7}, \frac{2}{7}, \frac{2}{7}, \frac{2}{7}\}$ and an example for $FFD = \frac{5}{9} + \frac{11}{9} OPT$. This might be the 'easy to construct' example Yue[72] refers to.

4) The ratio 7/6 is a lower bound for the asymptotic ratio of lists with no 1-items.

List $\mathcal{L} = \{ \underbrace{7,\ldots,7}_{6k+5},\ 5,\ \underbrace{4,\ldots,4}_{12k+10} \}$ on bins of size 15.

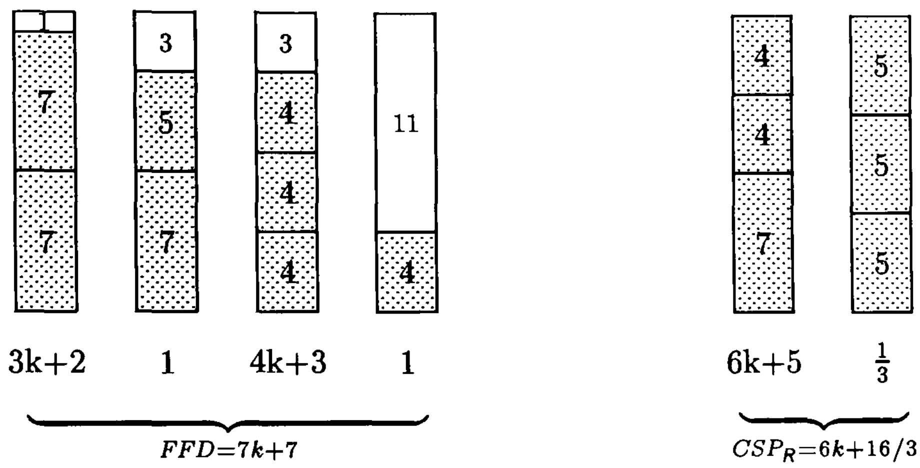


Diagram 8.9. Example for $FFD = \frac{7}{9} + \frac{7}{6} CSP_R$

List $\mathcal{L} = \{ \underbrace{3,\ldots,3}_{6k+2},\ \underbrace{2,\ldots,2}_{12k+4} \}$ on bins of size 7.

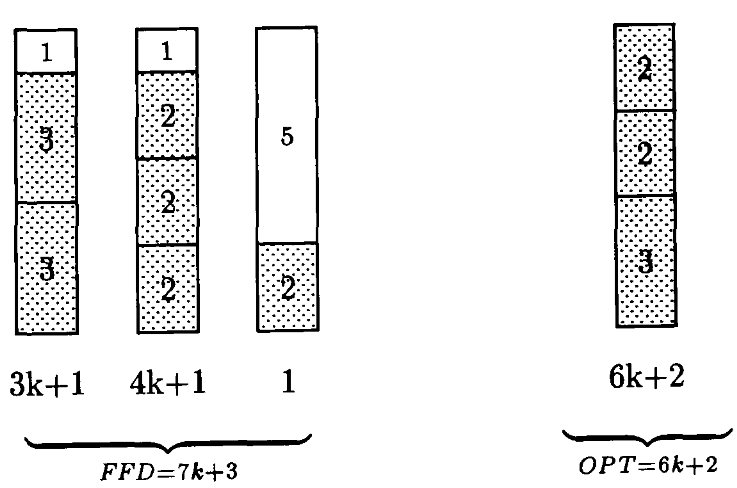


Diagram 8.10. Example for $FFD = \frac{2}{3} + \frac{7}{6} OPT$.

8.8 $x \in \langle \frac{1}{5}, \frac{1}{4}]$

We divide the interval $[x, 1-2x]$ into five sub-intervals as follows

$$[x, 1-2x]: \begin{cases} I_1 = \langle \frac{1}{2}, 1-2x], & \text{1-items} \\ I_2 = \langle \frac{1-x}{2}, \frac{1}{2}], & \text{2-items} \\ I_3 = \langle \frac{1}{3}, \frac{1-x}{2}], & \text{2-items} \\ I_4 = \langle \frac{1-x}{3}, \frac{1}{3}], & \text{3-items} \\ I_5 = [x, \frac{1-x}{3}], & \text{3 and 4-items} \end{cases} \tag{8.29}$$

Note that the condition $x < \frac{1}{4}$ is necessary when there 1-items in the list. Any list with critical item $x \in \langle \frac{1}{5}, \frac{1}{4}]$ packs into a configuration as shown in diagram 8.11.

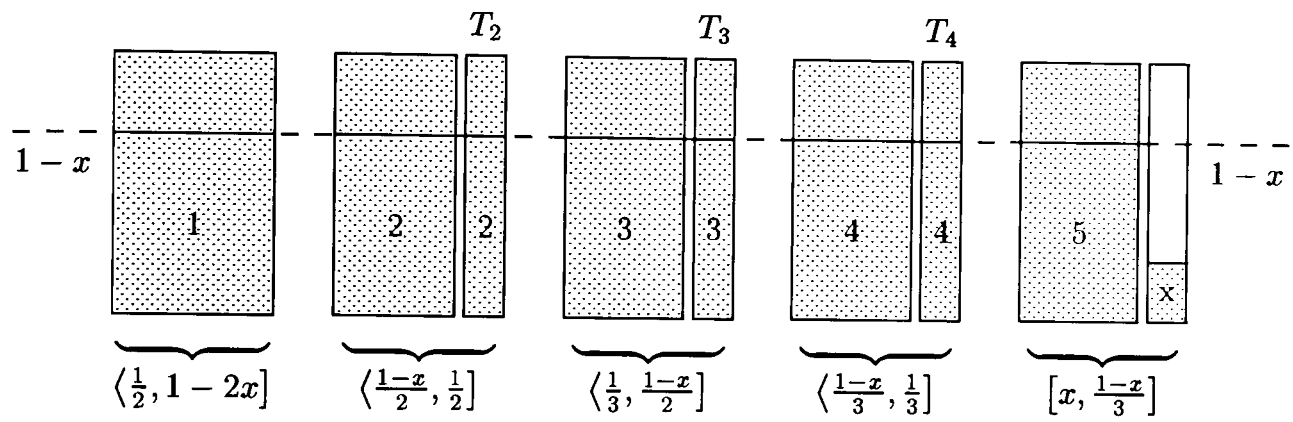


Diagram 8.11. General bin-configuration for $x \in \langle \frac{1}{5}, \frac{1}{4}]$

The diagram is to be read as follows. All bins in block 1, if there are any, have a largest item in the interval $I_1 = \langle \frac{1}{2}, 1-2x]$ and so on for blocks 2–5. When the last bin in block $i \in \{2, 3, 4\}$ does not contain the maximum number of items in the interval I_i we refer to it as a *transition bin.*[7] Note that a transition bin (T-bin) and the last bin are the only bins that can have a bin weight less than one under a recurrent weighting function.

We first derive some features of the minimal configuration, based upon the assumptions made in section 8.3. We then concentrate on the structure of blocks 1 and 2. It turns out that the structure of block 1 provides the key to the analysis of the case $\beta(x) = 4$. This leads to splitting this case into a further 6 subcases. Each of these subcases is then analysed in a fashion similar to the one used for the case $\beta(x) = 3$. However, for some cases we shall use a different method to derive an upper bound for the constant in the worst-case bound. This relies on formulating and solving a simple *set-packing problem* and is illustrated in section 8.8.2.

[7] This formulation differs slightly from the one by Baker,[3] but is in essence the same.

The derivation of the various weighting functions and most of the set-packing problems can be found in appendix D.1. This appendix also contains the balances necessary to construct the instances in the various diagrams.

8.8.1 Minimal configuration

We first apply the assumptions in section 8.3 to reduce the general configuration to a minimal one.

1. Block 5 consists of bins with $\beta = 4$ items of size x.
2. Blocks 1–4 consist of bins with at most 3 items by assumption 8.13. Further, since all items in block 3 and 4 are smaller than or equal to $(1-x)/2$ it follows that every bin in these blocks contains exactly 3 items.

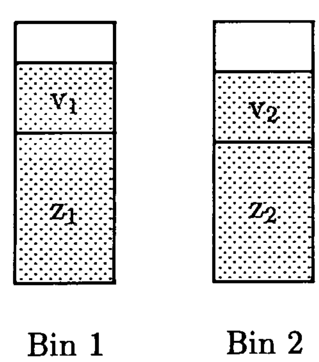
We now analyse the structure of blocks 1–4 in more detail.

Structure of 1-bins

Any 1-bin can contain at most 2 items. This follows from $\frac{1}{2} \geqslant 2x$ and assumption 8.13. By the same token a 1-bin cannot contain an additional item in the range $[2x, 1/2]$. Since all 1-items are $\leqslant 1-2x$ it follows that all 1-bins contain exactly 2 items, with the smallest item in the bin strictly less than $2x$.

$1 - 2x \geqslant z_1 \geqslant z_2 > \frac{1}{2}$

$2x > v_1 \geqslant v_2$



$z_1 + v_1 > 1 - x$

$z_2 + v_2 > 1 - x$

Diagram 8.12. Structure of first two 1-bins for $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$.

We now examine diagram 8.12 which shows the first two 1-bins. In the first bin, together with the largest 1-item (z_1), FFD will place the largest item that will fit (v_1). By assumption this item must be strictly less than $2x$, but since $z_1 \leqslant 1 - 2x$ it must be that v_1 is the first item in the list strictly less than $2x$. By the same logic, v_2 must be the second item in the list strictly less than $2x$ and thus v_1 and v_2 are consecutive items in the list. Since $z_1 + v_1 \geqslant z_2 + v_2 > 1 - x$ we can reduce the size of item $z_1 \to z_2$ and the size of item $v_1 \to v_2$ without altering the sequence of the items

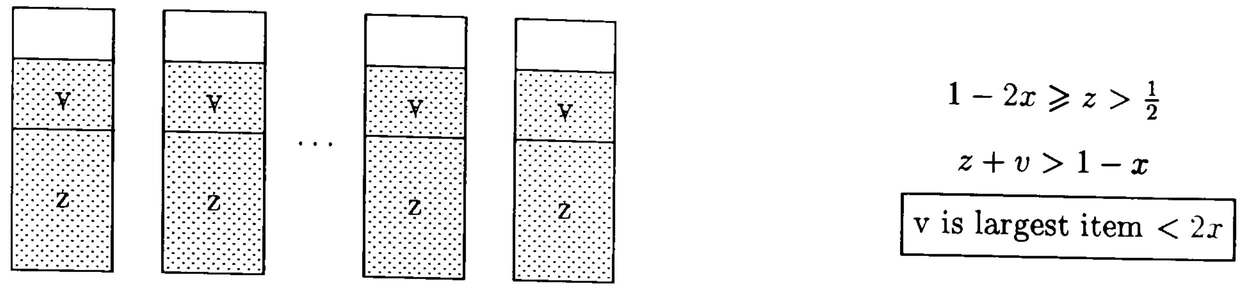


Diagram 8.13. Minimal 1-bin configuration for $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$

in the list or the number of bins that FFD uses. Ergo, we may assume that $z_1 = z_2$ and $v_1 = v_2$. Repeating this argument we conclude that all 1-bins are identical and have a structure as shown in diagram 8.13.

Structure of block 2

By assumptions 8.14 and 8.15 all the items in block 2 in the interval $I_2 = \langle \frac{1-x}{2}, \frac{1}{2}]$ have the same size, say w. This gives two possible configurations for block 2 as shown in diagram 8.14, depending on whether or not $w < 2x$ holds.

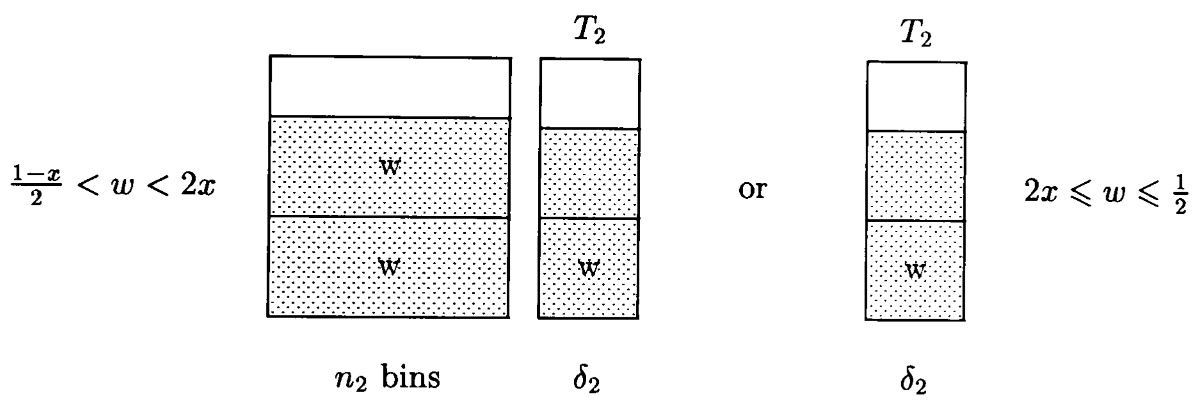
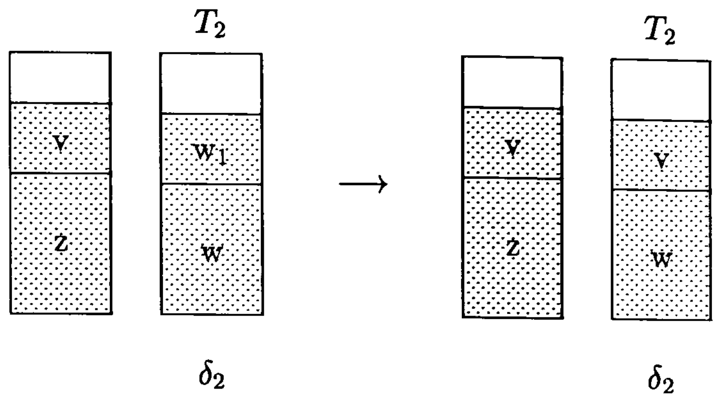


Diagram 8.14. Minimal bin-configuration of block 2 for $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$.

- If $w \geqslant 2x$ then $n_2 = 0$ and the transition bin, if there is one, can contain at most 2 items. This follows from assumption 8.13 (otherwise we can create a bin with 4 items of size x).
- If there are 1-items in the list, then $w > 2x$ must hold.

Assume that $w < 2x$ holds. This implies that $w \leqslant v$, since v is the largest item $< 2x$. But, if this is the case we can reduce both $z, v \to w$ and produce the same packing without 1-items. This gives a contradiction and thus $w \geqslant 2x$ must hold. If $w = 2x$ then FFD would have placed this item in a 1-bin, since $z \leqslant 1 - 2x$, but this contradicts the assumption that a 1-bin does not contain an item in $[2x, \frac{1}{2}]$. Ergo $w > 2x$.

- If there is a 1-item then block 2 can consist of only a T-bin with the largest item $w > 2x$. This



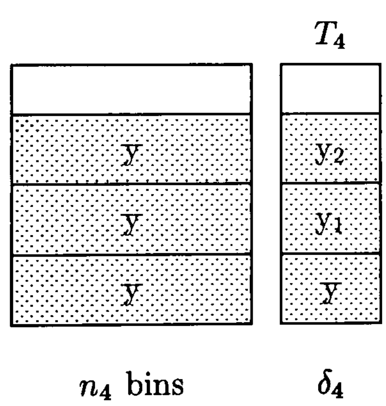
$$1 - 2x \geqslant z > \tfrac{1}{2}$$
$$z + v > 1 - x$$
$$\text{v is largest item} < 2x$$
$$2x < w \leqslant \tfrac{1}{2}$$
$$z + w > 1 \text{ and } w + v > 1 - x$$

Diagram 8.15. Structure of T_2 when list contains 1-items.

means that there can only be one other item (w_1) in this bin, which must be strictly smaller than $2x$ and thus $w_1 \leqslant v$ must hold. We can now reduce $v \to w_1$, since $w + w_1 > 1 - x$. This implies the structure as shown in diagram 8.15.

Structure of block 3 & 4

It turns out that the exact structure of block 3 is not important for a minimal configuration that achieves a worst-case bound. Block 4 has a structure similar to that of block 2. Apply assumptions 8.14 and 8.15 to arrive at diagram 8.16.



$$y \geqslant y_1 \geqslant y_2$$
$$\tfrac{1-x}{3} < y \leqslant \tfrac{1}{3}$$
$$y + y_1 + y_2 > 1 - x$$

Diagram 8.16. Structure of block 4 for $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$.

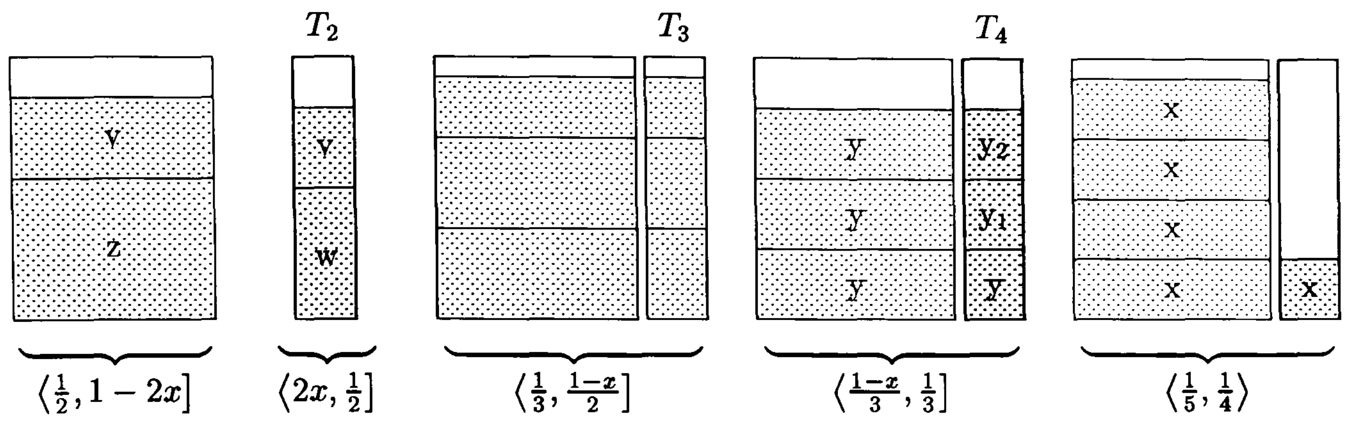


Diagram 8.17. Minimal configuration for list with 1-items and $x \in \langle\frac{1}{5}, \frac{1}{4}\rangle$.
The subscripted intervals indicate that the largest item in those bin(s) has a size in that interval.

This finally gives the structure as shown in diagram 8.17 for a minimal configuration, when there are 1-items in the list. In addition we can make the following assumptions for the size of the item v, for a list with 1-items.

8.25 Assumption $x < v \leqslant \frac{1-x}{2}$

Proof. *If $v > \frac{1-x}{2}$ we can reduce both $z, w \to v$ and FFD will produce a packing without 1-items that uses the same number of bins. So we may assume $v \leqslant \frac{1-x}{2}$. If $v = x$ then FFD will pack two items x in a 1-bin (since $z \leqslant 1 - 2x$) and we can replace a 1-bin by a bin with 4 items of size x (since $2x < z$). Ergo $v > x$.* □

8.26 Assumption *If $v > \frac{1}{2} - x$ then there is a bin with largest item in $\langle 2x, \frac{1}{2}]$.*

Proof. *If there is no transition bin on $\langle 2x, \frac{1}{2}]$ then we can reduce the last 1-item packed; $z \to \frac{1}{2}$. This will pack in the same number of bins since $\frac{1}{2} + v > 1 - x$. So we may assume that there is a transition bin on $\langle 2x, \frac{1}{2}]$.* □

8.27 Assumption *If $v \leqslant \frac{1}{2} - x$ then there is no bin with largest item in $\langle 2x, \frac{1}{2}]$, furthermore we may assume that z is the smallest size greater than $1 - x - v$.*

Proof. *Since $w + v \leqslant \frac{1}{2} + \frac{1}{2} - x = 1 - x$, FFD will place another item in this bin. But then we can replace this bin by a bin with 4 items of size x, so that we may assume that there is no such bin. This proves the first part of the assumption. The second part follows since $z > 1 - x - v \geqslant \frac{1}{2}$ and $z + v > 1 - x$ mean that FFD will produce the same packing when we reduce z to any size strictly greater than $1 - x - v$.* □

case	v-range	$W(z)$
1	$v \in \langle \frac{1}{3}, \frac{1-x}{2}]$	$\frac{7}{12}$
2	$v \in \langle \frac{1}{2} - x, \frac{1}{3}]$	$\frac{5}{8}$
3	$v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$	$\frac{23}{36}$
4	$v \in \langle \frac{1}{4}, \frac{1-x}{3}]$	$\frac{11}{16}$
5	$v \in \langle x, \frac{1}{4}]$	$\frac{7}{10}$
6	no 1-items	

Table 8.6. Cases for $\beta(x) = 4$
$W(z)$ is the weight used for the 1-item.

Ranges for v The configuration in diagram 8.17 is analysed by conditioning on the size of v. By assumption 8.25 we know that we only need to consider the range $\langle x, \frac{1-x}{2}]$. This range is broken down into 5 subcases as shown in table 8.6. This split was determined by taking the different ranges in the generic weighting function as a starting point. For every interval a bound was determined. When the ratio [in this bound] could not be proven to be tight, the interval was divided further. This process ended with the division given in table 8.6.

This leaves the special case when there are no 1-items in the list. To derive an asymptotically tight bound for this case one needs to further subdivide this case and condition on the existence of a bin with largest item in $\langle \frac{1-x}{3}, \frac{1}{3}]$.

Weighting function We use the weights $W(z) = V$ and $W(v) = 1 - V$ for the items in the 1-bins. For the other items we use [weights based upon] the generic weighting function. The parameter V is determined as to minimise the maximum pattern-weight (ratio).

8.8.2 $v \in \langle \frac{1}{3}, \frac{1-x}{2}]$ [8]

We start with the configuration as depicted in diagram 8.17. Since $v > \frac{1}{3} > \frac{1}{2} - x$, we can apply assumption 8.26 and get the following minimal configuration for this case.

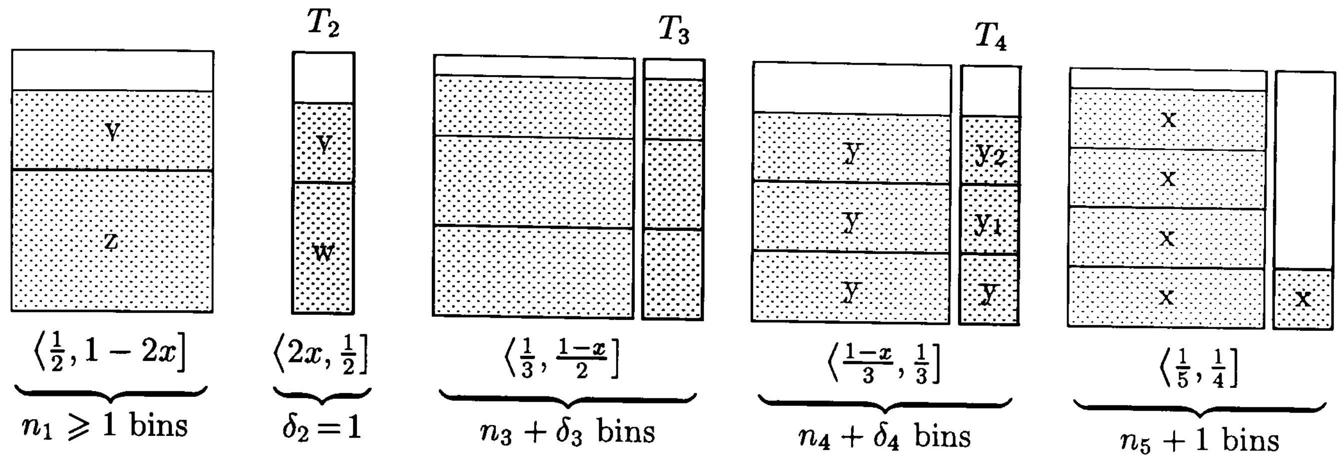


Diagram 8.18. Minimal configuration for list with 1-items, $v \in \langle \frac{1}{3}, \frac{1-x}{2}]$ and $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$

We use the following weights and weighting function,

$$W(z) = \frac{7}{12},\ W(w) = \frac{1}{2} \text{ and } W(s) = \begin{cases} \frac{5}{12}, & s \in \langle \frac{1}{3}, v] \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1}{3}] \\ \frac{1}{4}, & s \in [x, \frac{1-x}{3}] \end{cases} \tag{8.30}$$

With these weights and (5.7), we get the bound

$$FFD \leqslant \underbrace{(1 - \tfrac{11}{12}) + (1 - W_3)\,\delta_3 + (1 - W_4)\,\delta_4 + (1 - \tfrac{1}{4})}_{\text{constant c}} + W(\mathcal{L}), \tag{8.31}$$

where $\delta_{3,4} \in \{0, 1\}$ and $W_{3,4}$ is the weight of transition bins $T_{3,4}$ (if present).

Set-packing formulation We want to find a configuration of transition bins that maximises the constant c in (8.31). We can restrict ourselves to those configurations, where all T-bins have a weight strictly less than 1, otherwise we can set the corresponding δ_T to zero. Enumerating all possible transition patterns with weight strictly less than one, gives table 8.7 with 5 patterns. In this table each column represents different configurations of one transition bin. For instance, patterns 3 and 4 are possible configurations for bin T_4. Pattern 3 represents the situation where transition bin T_4 contains two items with size in $\langle \frac{1-x}{3}, \frac{1}{3}]$ and one item with size in $[x, \frac{1-x}{3}]$ to give a bin weight of $\frac{11}{12}$. Since each section in the table represents different configurations of the same transition bin we can select at most one pattern per section. In addition to this there are other

[8] Details of derivations can be found in appendix D.1.1 (p. 185)

		*			*	*
		1	2	3	4	5
$W_T \times 12$		11	11	11	10	3
$(1 - W_T) \times 12$		1	1	1	2	9
w	6/12	1	0	0	0	0
$\langle \frac{1}{3}, v]$	5/12	1	1	0	0	0
$\langle \frac{1-x}{3}, \frac{1}{3}]$	4/12	0	0	2	1	0
$[x, \frac{1-x}{3}]$	3/12	0	2	1	2	1

Table 8.7. Transition bins for configuration 8.18

constraints, induced by the sequence in which FFD packs the items into the bins. For example we cannot select both patterns 2 and 3. This is because FFD cannot have placed an item in $[x, \frac{1-x}{3}]$ in a bin with an item in $[\frac{1}{3}, v]$ if there is a subsequent bin with an item in $\langle \frac{1-x}{3}, \frac{1}{3}]$. The same holds for patterns 2 and 4. So selecting pattern 2 'knocks out' patterns 3 and 4. Given these restrictions, the choice of patterns from table 8.7 that maximises the constant c is patterns 1, 4 and 5 with a total value of $\frac{1}{12} + \frac{2}{12} + \frac{9}{12} = 1$. An upper bound for c is therefore 1, which gives $FFD \leqslant 1 + W(\mathcal{L})$.

	1	2	3	4	5
$(1 - W_T) \times 12$	1	1	1	2	9
	1	0	0	0	0
	0	1	1	1	0
	0	0	0	0	1

Table 8.8. Set-packing constraints for configuration 8.18

A condensed representation of what combinations of patterns are feasible is given in table 8.8. The 'profit' of choosing a pattern is given in the top row. The last three rows are the constraints. For instance, the second row indicates that we can choose at most one of the patterns 2, 3 or 4. It is not difficult to see that this constitutes a *set-packing problem* (SPP), which (on this scale) is easily solved. The format shown in both diagrams is chosen in subsequent sections.[9]

[9] Rather than giving two separate (directly related) tables we merge them together and blank the zero-entries in the set-packing part of the table. Furthermore, the column representing the singleton bin will be omitted from subsequent tables, since this pattern *has* to be chosen (by assumption 8.10) and therefore does not affect the SPP-formulation. For an example of the 'condensed' format, see table 8.9 on page 109.

Ratio problem The maximum pattern-weight under this weighting function is given by the following program. Note that any item in $\langle 2x, \frac{1}{2}]$ can be excluded from the formulation, since such an item is dominated.

$$r = \boxed{\begin{array}{ll} \text{Max} & \frac{7}{12}a_1 + \frac{5}{12}a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + x a_4 \leqslant 1 \\ & x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \text{ and } a_i \in \mathbb{N} \end{array}} \tag{8.32}$$

This program has a value of 7/6, so that $W(\mathcal{L}) \leqslant \frac{7}{6} CSP_R$. Combining this with the previous bound in $W(\mathcal{L})$ gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.23.

$$FFD \leqslant 1 + \tfrac{7}{6} CSP_R \tag{8.33}$$

8.8.3 $v \in \langle \frac{1}{2} - x, \frac{1}{3}]$

Since $v \leqslant \frac{1}{3}$ holds, the list does not contain any items in the interval $\langle \frac{1}{3}, \frac{1-x}{2}]$ and as a consequence the third block in diagram 8.17 is empty. This gives the following configuration.

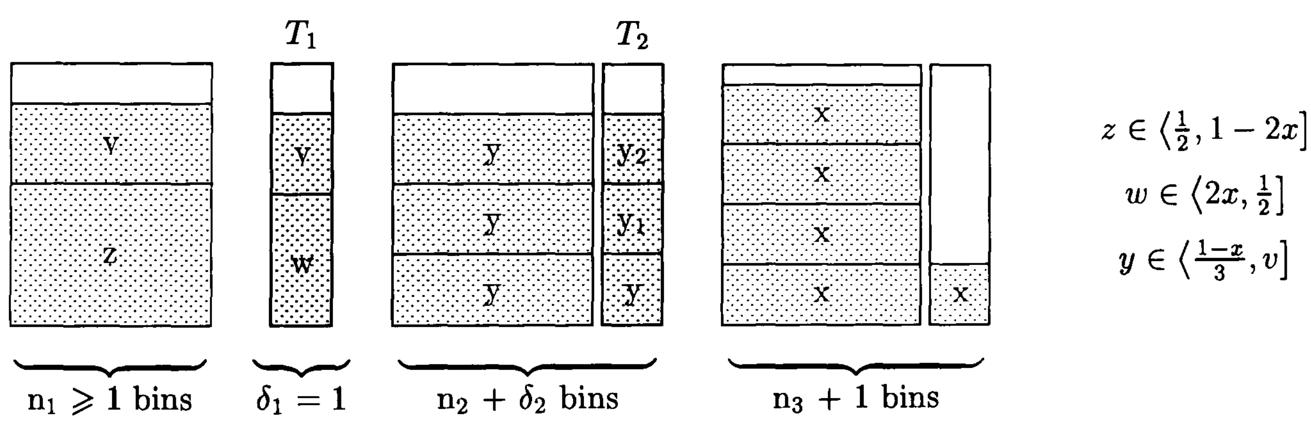


Diagram 8.19. Minimal configuration for list with 1-items, $v \in \langle \frac{1}{2} - x, \frac{1}{3}]$ and $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$.

We use the following weights for the items

$$W(z) = \tfrac{5}{8},\ W(w) = \tfrac{29}{48},\ W(v) = \tfrac{3}{8},\ W(y) = \tfrac{1}{3} \text{ and } W(y_1) = W(y_2) = W(x) = \tfrac{1}{4} \tag{8.34}$$

Applying (5.7) gives the bound $FFD \leqslant 15/16 + W(\mathcal{L})$. The maximum pattern-weight is determined in appendix D.1.2 (p. 187) as 29/24. Combining this gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.24.

$$FFD \leqslant \tfrac{15}{16} + \tfrac{29}{24} CSP_R \tag{8.35}$$

8.8.4 $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$

Since $v \leqslant \frac{1}{2} - x < \frac{1}{3}$, block 3 in diagram 8.17 must be empty. Furthermore, block 2 is empty since $w + v \leqslant 1 - x$ and FFD would have placed 3 items in this bin, which contradicts the assumption that this bin contains exactly two items. We therefore have the following configuration.

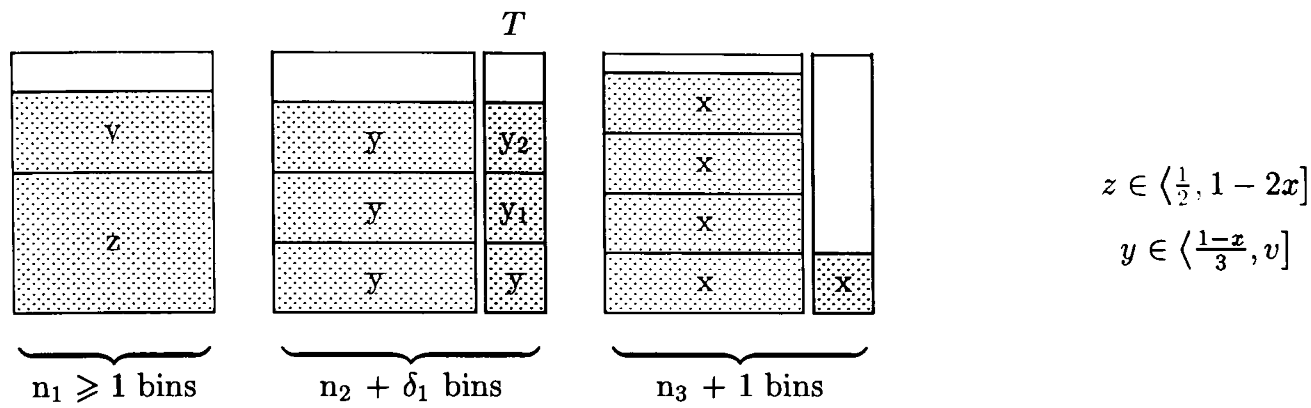


Diagram 8.20. Minimal configuration for list with 1-items, $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$ and $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$.

We use the following weights for the items

$$W(z) = \tfrac{23}{36},\ W(v) = \tfrac{13}{36},\ W(y) = \tfrac{1}{3},\ \text{and } W(y_1) = W(y_2) = W(x) = \tfrac{1}{4} \tag{8.36}$$

This gives the bound $FFD \leqslant 11/12 + W(\mathcal{L})$. The maximum pattern-weight is determined in appendix D.1.3 (p. 189) as 11/9. Combining this gives the following bound, which is asymptotically tight as shown by the instances in diagrams 8.25 and 8.26.

$$FFD \leqslant \tfrac{11}{12} + \tfrac{11}{9} CSP_R \tag{8.37}$$

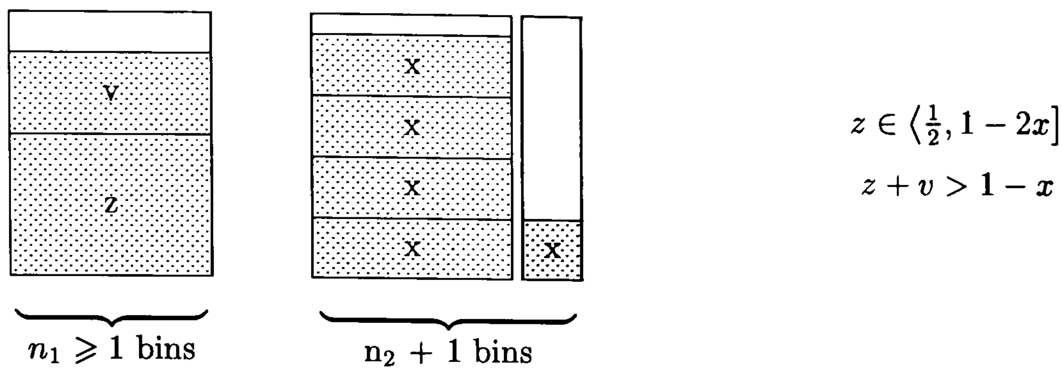


Diagram 8.21. Minimal configuration for $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $x < v \leqslant \frac{1-x}{3}$.

8.8.5 $v \in \langle \frac{1}{4}, \frac{1-x}{3}]$

We have a configuration as shown in diagram 8.21, and use the following weights for the items.

$$W(z) = \tfrac{11}{16},\ W(v) = \tfrac{5}{16} \text{ and } W(x) = \tfrac{1}{4} \tag{8.38}$$

This immediately gives $FFD = 3/4 + W(\mathcal{L})$. The maximum pattern-weight under these weights is determined in appendix D.1.4 (p. 189) as 19/16. Combining this gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.27.

$$FFD \leqslant \tfrac{3}{4} + \tfrac{19}{16} CSP_R \tag{8.39}$$

8.8.6 $v \in \langle x, \frac{1}{4}]$

We have a configuration as shown in diagram 8.21, and use the following weights for the items.

$$W(z) = \tfrac{7}{10},\ W(v) = \tfrac{3}{10} \text{ and } W(x) = \tfrac{1}{4} \tag{8.40}$$

This immediately gives $FFD = 3/4 + W(\mathcal{L})$. The maximum pattern-weight under these weights is determined in appendix D.1.5 (p. 192) as 6/5. Combining this gives the bound $FFD \leqslant 3/4 + (6/5)CSP_R$, and implies the bound $FFD \leqslant 3/5 + (6/5)OPT$.

The bound in CSP_R is sufficient to prove (8.1) and we could stop here. However, we will take the opportunity to demonstrate that tight bounds cannot always be obtained by a weighting function alone. Although the bound in OPT is tight, as is illustrated by diagram 8.29, the bound in CSP_R is not. To reduce the constant we need to do some further analysis. We note that $\mathbf{u}^\mathsf{T} = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$ and $\mathbf{u}^\mathsf{T} = [\frac{3}{4}, \frac{1}{4}, \frac{1}{8}]$ are both valid dual multipliers. So that $CSP_R \geqslant \max\{\frac{3}{4}n_1 + n_2 + \frac{1}{4}, n_1 + \frac{1}{2}n_2 + \frac{1}{8}\}$ and thus $FFD - 6/5\,CSP_R \leqslant \min\{\frac{7}{10} + \frac{1}{10}(n_1 - 2n_2), \frac{17}{20} - \frac{1}{5}(n_1 - 2n_2)\} = 7/10$. This gives the following bound, which is tight as shown by the instance in diagram 8.28.

$$FFD \leqslant \tfrac{7}{10} + \tfrac{6}{5} CSP_R \tag{8.41}$$

8.8.7 No 1-items

We have the following configuration.

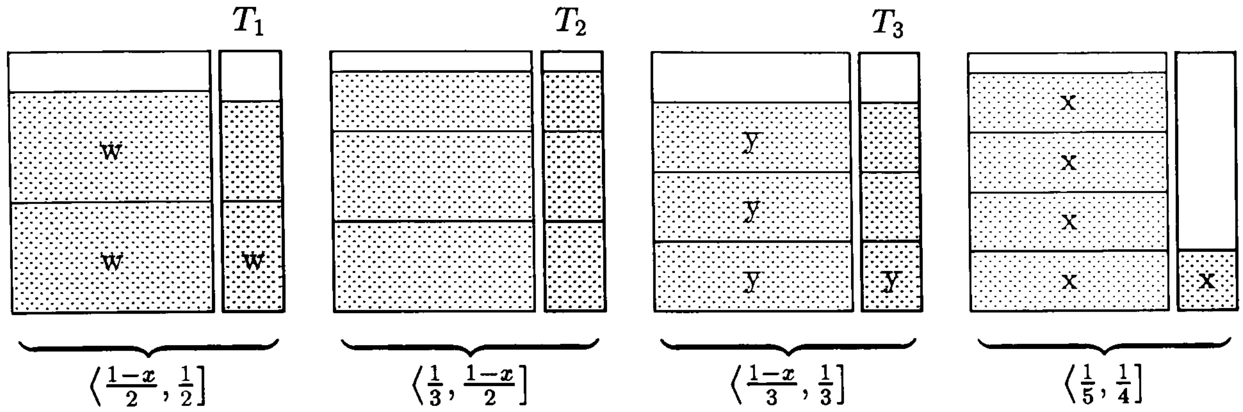


Diagram 8.22. General configuration for list without 1-items and $x \in \langle \frac{1}{5}, \frac{1}{4}]$.

We can use the generic weighting function with $W_x = \frac{4}{15}$ to prove the bound $FFD \leqslant \frac{31}{30} + \frac{6}{5} CSP_R$. This bound is sufficient to prove (8.1). However, it is known[37] that the asymptotic ratio for lists with no 1-items is $\frac{71}{60}$, so that this bound cannot be tight. Furthermore, this bound also admits the cases $(7, 5)$ and $(13, 10)$ for (FFD, OPT) (see table 8.2). By conditioning on the existence of y, which gives two cases to investigate, we derive the following bound.

$$FFD \leqslant \tfrac{13}{12} + \tfrac{7}{6} CSP_R \tag{8.42}$$

8.8.7a There is a bin with largest item $y \in \langle \frac{1-x}{3}, \frac{1}{3}]$

For this case we use the following variant of the generic weighting function.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \langle \frac{1-x}{2}, \frac{1}{2}] \\ \frac{5}{12}, & s \in \langle \frac{1-y}{2}, \frac{1-x}{2}] \\ \frac{1}{3}, & s \in [y, \frac{1-y}{2}] \\ \frac{1}{4}, & s \in [x, y\rangle \end{cases} \tag{8.43}$$

By assumption 8.16 we know that there are no bins with largest item in $\langle \frac{1}{3}, \frac{1-y}{2}]$ and with this it is easy to verify that all recurrent bins have a bin weight of 1. The maximum pattern-weight under (8.43) is determined in appendix D.1.6a (p. 194) as 7/6.

To determine an upper bound for the constant we have listed all possible configurations for the transition bins with weight strictly less than one in table 8.9. Note that since we have conditioned on a bin with item y, the bin configurations 3 and 4 cannot represent a valid bin. For example,

		1	2	3	4	5	6
			*				*
$W_T \times 12$		11	10	9	11	11	10
$(1-W_T) \times 12$		1	2	–	–	1	2
$\langle \frac{1-x}{2}, \frac{1}{2}]$	6/12	1	1	1	0	0	0
$\langle \frac{1-y}{2}, \frac{1-x}{2}]$	5/12	1	0	0	1	0	0
$[y, \frac{1-y}{2}]$	4/12	0	1	0	0	2	1
$[x, y\rangle$	3/12	0	0	1	2	1	2
		1	1				
						1	1

Table 8.9. SPP-formulation for case a

		1	2	3
		*		*
$W_T \times 8$		7	6	7
$(1-W_T) \times 8$		1	2	1
$\langle \frac{1-x}{2}, \frac{1}{2}]$	4/8	1	1	0
$\langle \frac{1}{3}, \frac{1-x}{2}]$	3/8	1	0	1
$[x, \frac{1}{3}]$	2/8	0	1	2
		1	1	
			1	1

Table 8.10. case b

consider pattern 3; after placing the first item the bin is filled to a level $\leqslant \frac{1}{2} < 1 - y$, and FFD would have placed an item y in this bin. The same holds for pattern 4.

The set-packing problem gives a value of 4/12 for pattern combination (2, 6). This implies an upper bound for the constant of $\frac{4}{12} + 1 - \frac{1}{4} = \frac{13}{12}$. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.30.

$$FFD \leqslant \tfrac{13}{12} + \tfrac{7}{6} CSP_R \tag{8.44}$$

8.8.7b There is no bin with largest item $y \in \langle \frac{1-x}{3}, \frac{1}{3}]$

For this case we use the following variant of the generic weighting function.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \langle \frac{1-x}{2}, \frac{1}{2}] \\ \frac{3}{8}, & s \in \langle \frac{1}{3}, \frac{1-x}{2}] \\ \frac{1}{4}, & s \in [x, \frac{1}{3}] \end{cases} \tag{8.45}$$

The maximum pattern-weight under (8.45) is determined in appendix D.1.6b (p. 194) as 9/8. A bound for the constant is easily derived from table 8.10.

The set-packing problem in table 8.10 gives a value of 2/8 for pattern combination (1, 3), so that an upper bound for the constant is $\frac{2}{8} + 1 - \frac{1}{4} = 1$. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.31.

$$FFD \leqslant 1 + \tfrac{9}{8} CSP_R \tag{8.46}$$

8.8.8 Overview of bounds for $x \in \langle \frac{1}{5}, \frac{1}{4}]$

	Case	Bound	c	r	$\frac{8}{9}$	$\frac{7}{9}$	
1	1-item, $v \in \langle \frac{1}{3}, \frac{1-x}{2}]$	$FFD \leqslant 1 + \frac{7}{6}\, CSP_R$		*	—	—	√
2	1-item, $v \in \langle \frac{1}{2} - x, \frac{1}{3}]$	$FFD \leqslant \frac{15}{16} + \frac{29}{24} CSP_R$		*	—	(13,10)	√
3	1-item, $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$	$FFD \leqslant \frac{11}{12} + \frac{11}{9} CSP_R$		*	n.a.		√
4	1-item, $v \in \langle \frac{1}{4}, \frac{1-x}{3}]$	$FFD \leqslant \frac{3}{4} + \frac{19}{16} CSP_R$	*	*	—	—	√
5	1-item, $v \in \langle x, \frac{1}{4}]$	$FFD \leqslant \frac{7}{10} + \frac{6}{5}\, CSP_R$	*	*	—	—	√
6	no 1-item	$FFD \leqslant \frac{13}{12} + \frac{7}{6}\, CSP_R$		*	—	—	√

Table 8.11. FFD-bounds for $\beta(x) = 4$
A tick indicates that for this case bound (8.1) holds

The bounds for the various subcases, derived in the previous sections, are summarised in table 8.11. An asterisk in the 'ratio' column indicates that this bound is asymptotically tight. An asterisk in the 'constant' column indicates that there is an instance for which this bound holds with equality. Diagrams with the various instances can be found on pages 112–114.

Our primary aim is to prove that bound (8.1) holds for $x \in \langle 1/5, 1/4]$. From table 8.11 it is obvious that it holds for cases 1–5. For case 6 we can use corollary 8.19, which implies that we only need to check for $CSP_R > 3\frac{1}{4}$. A straightforward comparison will show that (8.1) is also satisfied for this case. This proves, taking into account assumption 8.22, the following bound.

$$x \in \langle \tfrac{1}{5}, \tfrac{1}{4}] \;\Rightarrow\; FFD < 1 + \tfrac{11}{9} CSP_R \tag{8.47}$$

A secondary objective is to draw some conclusions on the bound in OPT. The maximum value c^* of $c(\mathcal{L}) = FFD(\mathcal{L}) - \frac{11}{9} OPT(\mathcal{L})$ is one of $\{6/9, 7/9, 8/9\}$. This follows from (8.47) and the instance in diagram 8.26. To test which of these values is feasible we can use table 8.2 (p. 88). This shows that, excluding case 3 for the time being, none of the cases allows a value of 8/9 for c^*. Similarly we can show, again excluding case 3, that $(FFD, OPT) = (13, 10)$ is the only possibility for a value of 7/9 for c^*. Subsequent analysis in section D.1.7 shows that the bound $FFD \leqslant 7/8 + (29/24)OPT$ holds for case 2, which eliminates $(13, 10)$ as a possibility. It is also shown in section D.1.7 that the bound for case 3 can be tightened to $FFD \leqslant 62/81 + (11/9)CSP_R$, which rules out the values 7/9 and 8/9 for c^* for this case.

Combining all this proves the following bound, which is tight.

$$x \in \langle \tfrac{1}{5}, \tfrac{1}{4}] \;\Rightarrow\; FFD \leqslant \tfrac{2}{3} + \tfrac{11}{9} OPT \tag{8.48}$$

8.8.9 Comments

The diagrams on the following pages can be used to infer the following.

1) The 6/9-example in diagram 8.26 corresponds to the example (see diagram 8.2) that Johnson gives to achieve the 11/9-ratio. The sizes $\frac{1}{2}+\varepsilon$, $\frac{1}{4}+2\varepsilon$, $\frac{1}{4}+\varepsilon$, $\frac{1}{4}-2\varepsilon$ on bin size 1, (ε sufficiently small) generate the same extremal-pattern set as in table D.5. Moreover, if we choose ε such that all FFD-bins (in diagram 8.2) have the same wastage; viz. $1/4-3\varepsilon = 8\varepsilon \Rightarrow \varepsilon = 1/44$, we get the sizes as used in diagrams 8.25 and 8.26.

2) Note that we can reduce the singleton-bin item in diagrams 8.25 and 8.26 to any size in $\langle 8, 9]$, so that we can easily construct an example such that $x \in \langle \frac{2}{11}, \frac{9}{44}]$ and $FFD = \frac{2}{3} + \frac{11}{9} OPT$. This provides us with a lower bound for the case $\beta(x) = 5$.

3) Taking $k = 0$ in diagram 8.24 gives diagram 8.1 and an example for $FFD = \frac{2}{3} + \frac{11}{9} OPT$.

4) Using the bound $FFD \leqslant \lfloor \frac{15}{16} + \frac{29}{24} CSP_R \rfloor$ for case 3, alongside the other bounds in table 8.11, one can prove the following bound by direct comparison.

$$x \in \langle \tfrac{1}{5}, \tfrac{1}{4}] \text{ and } CSP_R \geqslant 11\tfrac{64}{99} \;\Rightarrow\; FFD \leqslant \tfrac{62}{81} + \tfrac{11}{9} CSP_R$$

This bound is tight as shown by the examples in diagrams D.8 and D.9 on page 201.

List $\mathcal{L} = \{\, \underbrace{15,\ldots,15}_{4k},\ \underbrace{10,\ldots,10}_{4k},\ \underbrace{8,\ldots,8}_{6k+9},\ \underbrace{6,\ldots,6}_{4k+9} \,\}$ on bins of size 29.

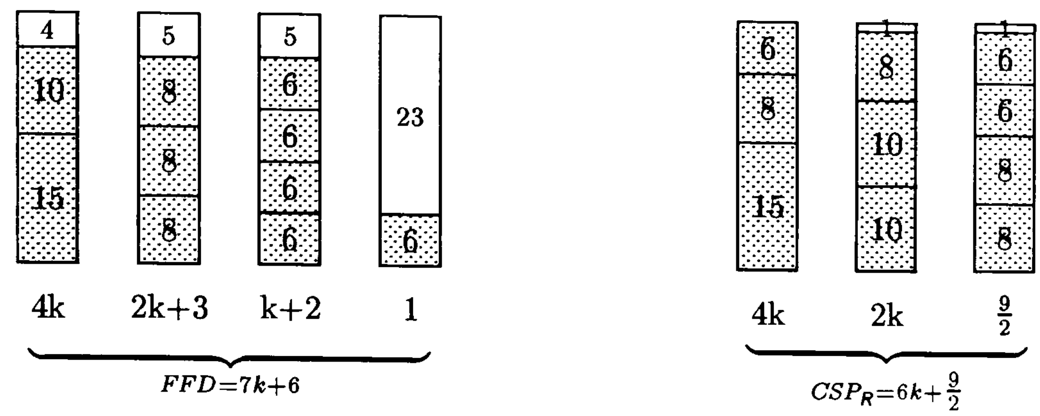


Diagram 8.23. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1}{3}, \frac{1-x}{2}]$; instance with $FFD{=}\frac{3}{4}{+}\frac{7}{6}\,CSP_R$.

List $\mathcal{L} = \{\, \underbrace{15,\ldots,15}_{12k+3},\ \underbrace{9,\ldots,9}_{12k+3},\ \underbrace{8,\ldots,8}_{24k+6},\ \underbrace{6,\ldots,6}_{36k+9} \,\}$ on bins of size 29.

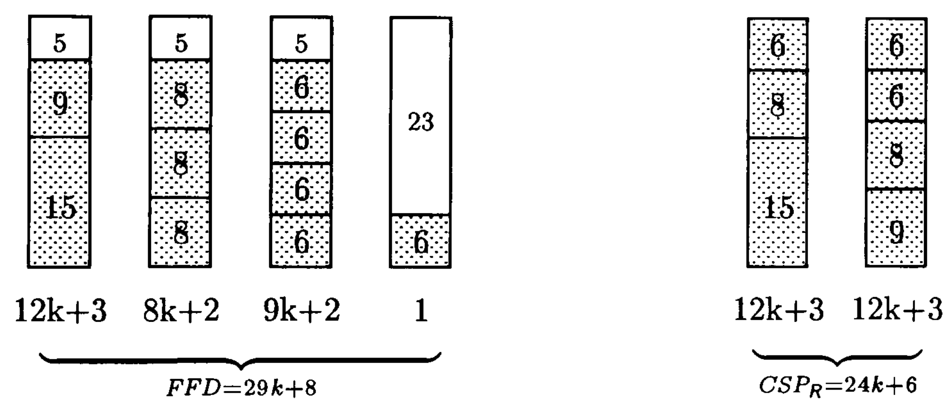


Diagram 8.24. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1}{2} - x, \frac{1}{3}]$; instance with $FFD{=}\frac{3}{4}{+}\frac{29}{24}\,CSP_R$.

List $\mathcal{L} = \{\, \underbrace{23,\ldots,23}_{6k+2},\ \underbrace{13,\ldots,13}_{6k+2},\ \underbrace{12,\ldots,12}_{6k+3},\ \underbrace{9,\ldots,9}_{12k+5} \,\}$ on bins of size 44.

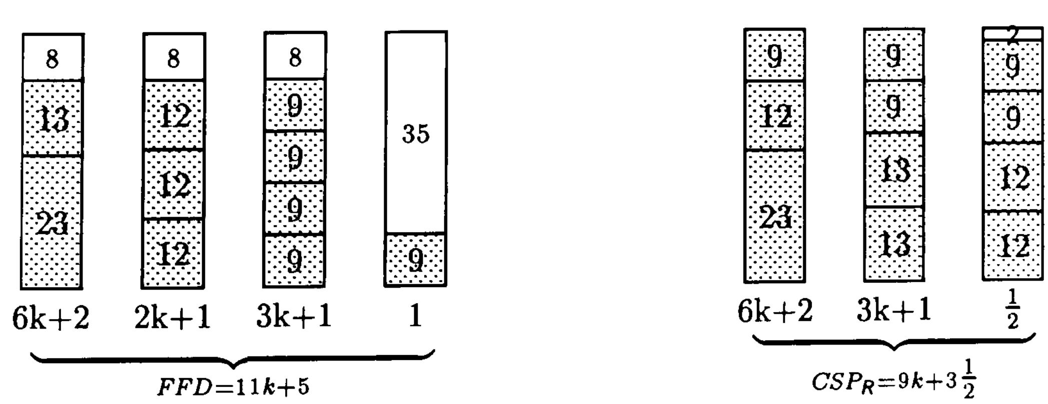


Diagram 8.25. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$; instance with $FFD{=}\frac{13}{18}{+}\frac{11}{9}\,CSP_R$.

List $\mathcal{L} = \{ \underbrace{23,\ldots,23}_{6k+4}, \underbrace{13,\ldots,13}_{6k+4}, \underbrace{12,\ldots,12}_{6k+3}, \underbrace{9,\ldots,9}_{12k+9} \}$ on bins of size 44.

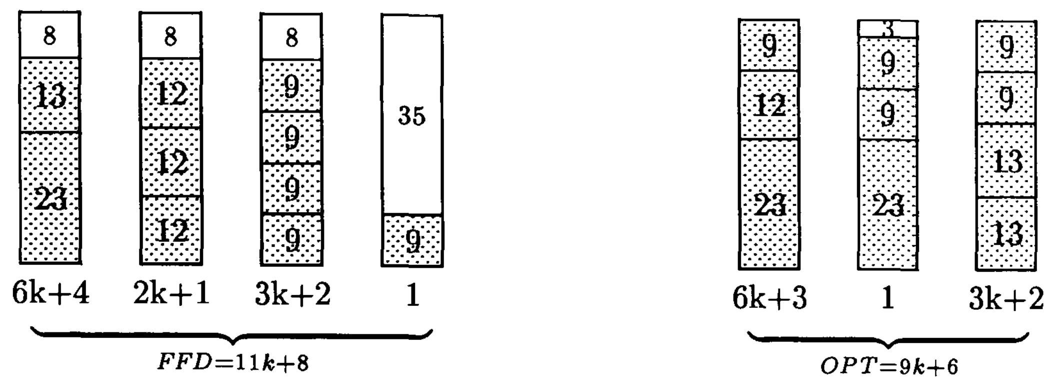


Diagram 8.26. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$; instance with $FFD = \frac{6}{9} + \frac{11}{9} OPT$.

List $\mathcal{L} = \{ \underbrace{11,\ldots,11}_{12k+9}, \underbrace{5,\ldots,5}_{12k+9}, \underbrace{4,\ldots,4}_{28k+21} \}$ on bins of size 19.

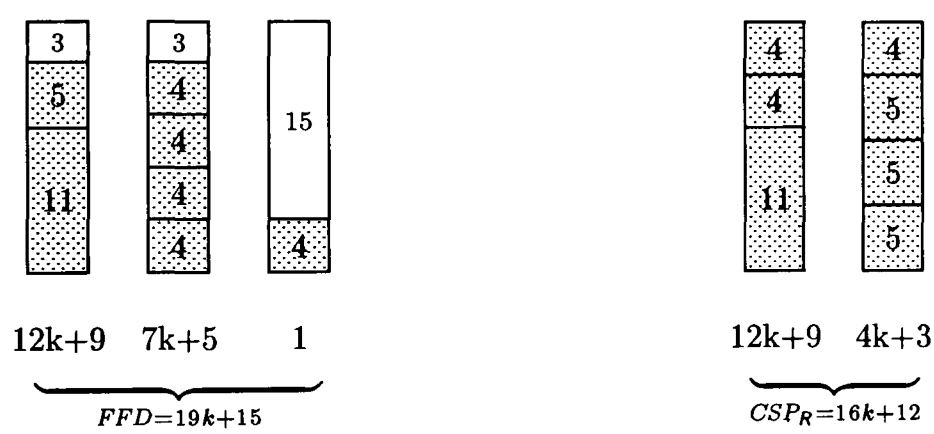


Diagram 8.27. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1}{4}, \frac{1-x}{3}]$; instance with $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$.

List $\mathcal{L} = \{ \underbrace{14,\ldots,14}_{2k}, \underbrace{6,\ldots,6}_{2k}, \underbrace{5,\ldots,5}_{4k+1} \}$ on bins of size 24.

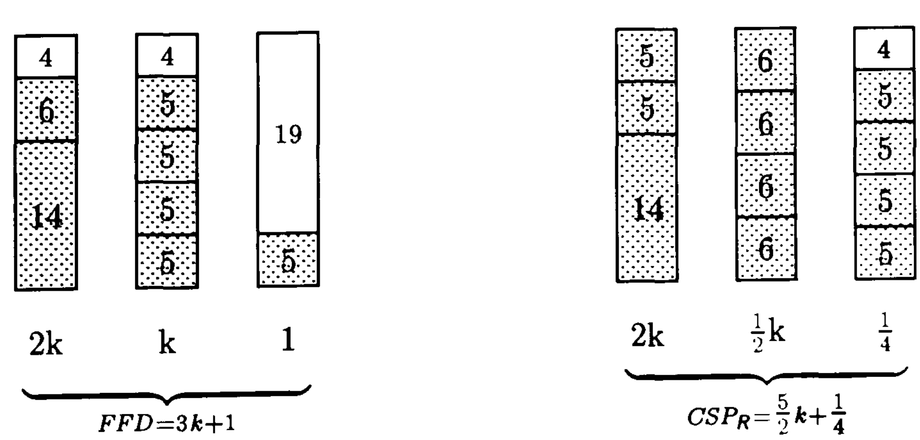


Diagram 8.28. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle x, \frac{1}{4}]$; instance with $FFD = \frac{7}{10} + \frac{6}{5} CSP_R$.

List $\mathcal{L} = \{ \underbrace{14,\dots,14}_{4k+9}, \underbrace{6,\dots,6}_{4k+9}, \underbrace{5,\dots,5}_{8k+21} \}$ on bins of size 24.

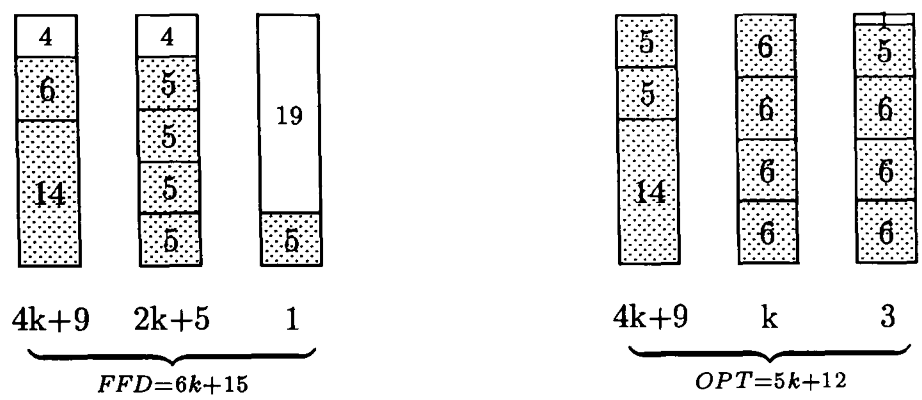


Diagram 8.29. $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle x, \frac{1}{4}]$; instance with $FFD = \frac{3}{5} + \frac{6}{5} OPT$

List $\mathcal{L} = \{ \underbrace{4,\dots,4}_{12k+3}, \underbrace{3,\dots,3}_{12k+5} \}$ on bins of size 14.

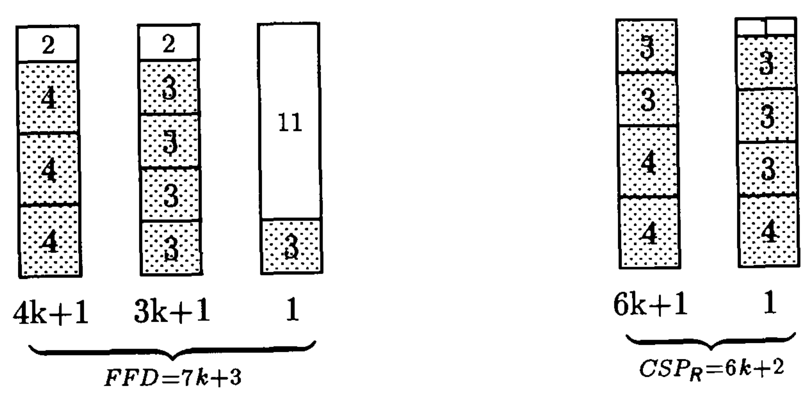


Diagram 8.30. $x \in \langle \frac{1}{5}, \frac{1}{4}]$ and no 1-item, case a; instance with $FFD = \frac{2}{3} + \frac{7}{6} CSP_R$.

List $\mathcal{L} = \{ \underbrace{5,\dots,5}_{8k+2}, \underbrace{3,\dots,3}_{24k+6} \}$ on bins of size 14.

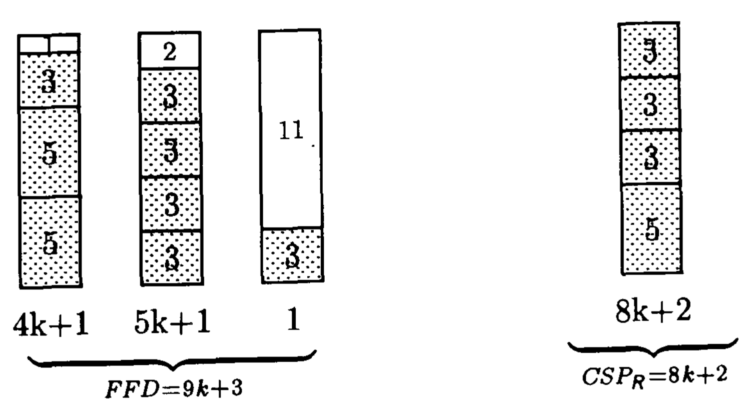


Diagram 8.31. $x \in \langle \frac{1}{5}, \frac{1}{4}]$ and no 1-item, case b; instance with $FFD = \frac{3}{4} + \frac{9}{8} CSP_R$.

8.9 Case $x \in \langle \frac{2}{11}, \frac{1}{5}]$

We divide the interval $[x, 1-2x]$ into eight sub-intervals as follows

$$[x, 1-2x] : \begin{cases} I_1 = \langle 3x, 1-2x], & \text{1-items} \\ I_2 = \langle \frac{1}{2}, 3x], & \text{1-items} \\ I_3 = \langle \frac{1-x}{2}, \frac{1}{2}], & \text{2-items} \\ I_4 = \langle \frac{1}{3}, \frac{1-x}{2}], & \text{2-items} \\ I_5 = \langle \frac{1-x}{3}, \frac{1}{3}], & \text{3-items} \\ I_6 = \langle \frac{1}{4}, \frac{1-x}{3}], & \text{3-items} \\ I_7 = \langle \frac{1-x}{4}, \frac{1}{4}], & \text{4-items} \\ I_8 = [x, \frac{1-x}{4}], & \text{4 and 5-items} \end{cases} \tag{8.49}$$

Note that the condition $x < \frac{1}{5}$ is necessary when there are 1-items in the interval I_1. Any list with critical item $x \in \langle \frac{2}{11}, \frac{1}{5}]$ packs into a configuration as shown in diagram 8.32.

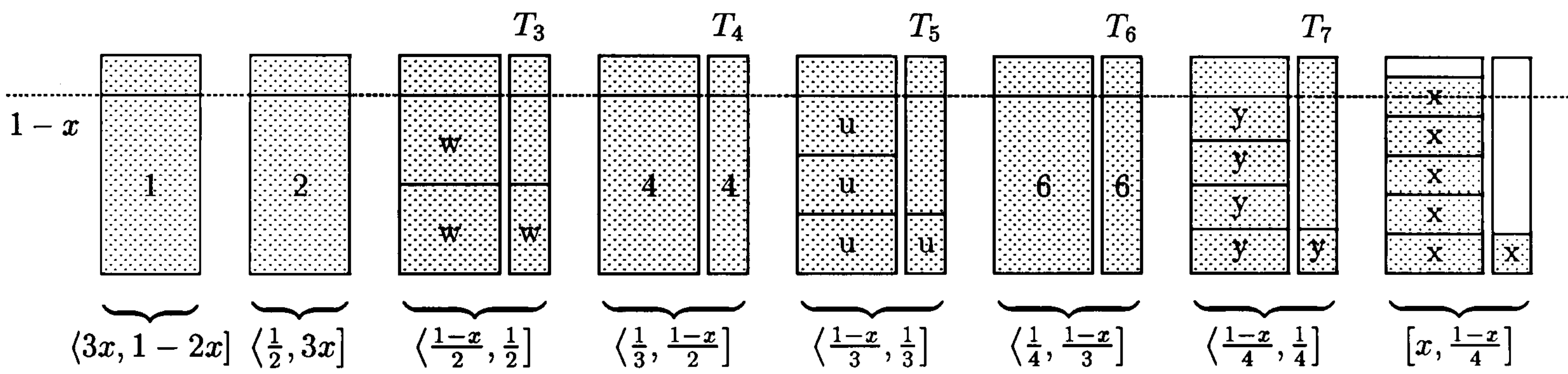


Diagram 8.32. General bin-configuration for critical item $x \in \langle \frac{2}{11}, \frac{1}{5}]$.

The diagram is to be read as follows. All bins in block 1, if there are any, have a largest item in the interval $I_1 = \langle 3x, 1-2x]$ and so on for blocks 2–8. When the last bin in block $i \in \{3, \ldots, 7\}$ does not contain the maximum number of items in the interval I_i it is referred to as a transition bin.

8.9.1 Minimal configuration

The structure of blocks 3, 5, 7 and 8 follows from the assumptions made in section 8.3.

1. Block 8 consists of bins with $\beta = 5$ items of size x.
2. Blocks 6 and 7 consist of bins with exactly 4 items. Moreover, we can apply assumptions 8.14 and 8.15 to block 7.

3. Blocks 4 and 5 consist of bins with exactly 3 items. Moreover, we can apply assumptions 8.14 and 8.15 to block 5.

4. Block 3 consists of bins with exactly 2 items. Again we can apply assumptions 8.14 and 8.15 to this block.

It turns out that the exact structure of blocks 4 and 6 is not important in the subsequent analysis. The structure of blocks 1 and 2 is analysed in more detail in the following sections.

8.9.1.1 Structure of 1-bins when there is a 1-item in $\langle 3x, 1-2x]$

• Any 1-bin with a 1-item in $\langle 3x, 1-2x]$ can contain at most 2 items. This follows from assumption 8.13, otherwise we can create a bin with 5 items of size x. For the same reason such a 1-bin cannot contain an item in the range $[2x, \frac{1}{2}]$.

• The largest item smaller than $2x$ will be placed in the first bin, with a 1-item in $\langle 3x, 1-2x]$. We can follow exactly the same analysis as for the 1-bins in the case of $x \in \langle \frac{1}{5}, \frac{1}{4}]$ and arrive at the following minimal configuration for block 1.

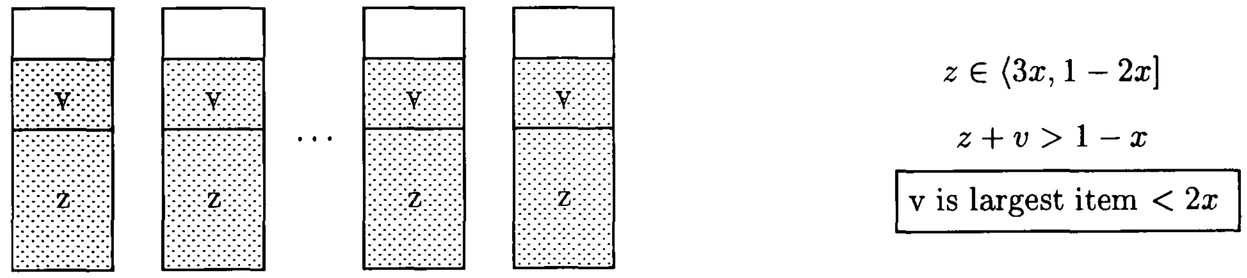


Diagram 8.33. Minimal 1-bin configuration for $z > 3x$ and $x \in \langle \frac{2}{11}, \frac{1}{5} \rangle$.

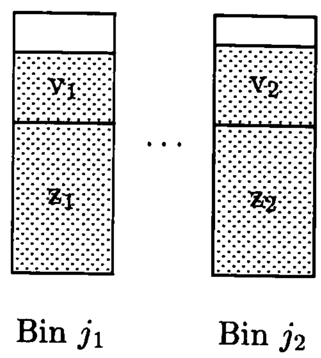
The structure of block 2 now follows from the following lemma.

8.28 Lemma *For a minimal configuration with 1-items in $\langle 3x, 1-2x]$ there are two possibilities for a bin with a 1-item in $\langle \frac{1}{2}, 3x]$. Either a bin contains two additional items, or a bin contains just one additional item and this item is strictly larger than $2x$.*

Proof. *Denote by z, v the items in bin 1 and let z' be the 1-item in a bin in block 2. That this bin cannot contain three additional items is obvious, since $\frac{1}{2} + 3x > 1$. Further since $z' \leqslant 3x$ there must be at least one additional item in the bin. This leaves the second case to be investigated. Suppose that the bin contains one additional item $v' \leqslant 2x$. If $v' = 2x$ then FFD would have placed this item in the first bin, so that $v' < 2x$. Since v is the largest item strictly less than $2x$ it follows* *that $v \geqslant v'$. But then we can reduce $z \rightarrow z'$ and $v \rightarrow v'$ to give a list which packs into the same number of bins, without any 1-items in $\langle 3x, 1-2x]$. Ergo $v' > 2x$ which proves the second case.* □

8.9.1.2 Structure of 1-bins when there are no 1-items in $\langle 3x, 1-2x]$

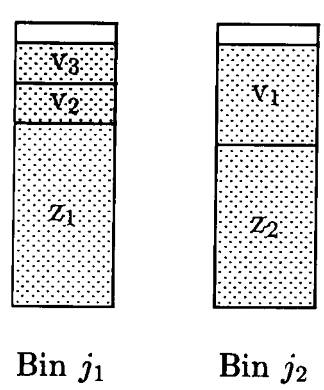
The structure of 1-bins with a 1-item in $\langle \frac{1}{2}, 3x]$ is slightly more complicated (block 1 of course is empty). It must contain at least two items since $3x \leqslant 1-2x$. It cannot contain four or more items since $\frac{1}{2} + 3x > 1$, so that such a bin must contain either 2 or 3 items.



$$v_{1,2} \leqslant 1-3x \quad \Rightarrow \quad v_1 = v_2 \text{ and } z_1 = z_2$$
$$v_1 > 1-3x \quad \Rightarrow \quad v_2 > 1-3x$$

Diagram 8.34. Structure of 1-bins in I_2 with one additional item

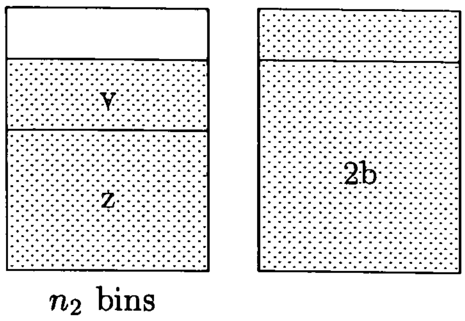
- Suppose we have a configuration as shown in diagram 8.34, where bins $j_{1,2}$ represent 1-bins with one additional item. If $v_{1,2} \leqslant 1-3x$ then $v_1 \geqslant v_2$ must hold, since $z_{1,2} \leqslant 3x$. But this means that we can reduce $v_1 \rightarrow v_2$ and $z_1 \rightarrow z_2$, without affecting the packing. If $v_1 > 1-3x$ and $v_2 \leqslant 1-3x$ then we can apply the same reduction. So that $v_1 > 1-3x$ implies $v_2 > 1-3x$.



$$\left.\begin{array}{c} 3x \geqslant z_1 \geqslant z_2 > \frac{1}{2} \\ z_1 + v_2 + v_3 > 1-x \\ z_2 + v_1 > 1-x \end{array}\right\} \Rightarrow v_1 > 1-3x$$

Diagram 8.35. Structure of 1-bins in I_2, containing two items, for $x \in \langle \frac{2}{11}, \frac{1}{5}]$

- Now suppose that we have a 1-bin with two additional items. This has consequences for the size of the items in subsequent 1-bins with one additional item. Consider the configuration shown in diagram 8.35. If $v_1 \leqslant v_2$ then $z_2 + v_1 \leqslant z_1 + v_2 \leqslant 1 - v_3 \leqslant 1-x$. So FFD would have placed another item in bin j_2. This gives a contradiction and thus $v_1 > v_2$ must hold. From $v_1 > v_2$ and the FFD-rule it follows that $z_1 + v_1 > 1$ must hold. This gives the lower bound $v_1 > 1 - z_1 \geqslant 1-3x$.



Block 2a: $\frac{1}{2} < z \leqslant 3x$, $\boxed{v \text{ is largest item} \leqslant 1 - 3x}$

Block 2b: 1-bins with two items, smallest item $> 1 - 3x$

1-bins with three items

Diagram 8.36. Structure of 1-bins with 1-item $\leqslant 3x$ and $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right]$.

We can combine all this in the following lemma.

8.29 Lemma *A minimal configuration with only 1-items in $\left\langle \frac{1}{2}, 3x \right]$ has a structure as in diagram 8.36*

Proof. *Condition on the structure of the first bin with a 1-item in $\left\langle \frac{1}{2}, 3x \right]$.*

1) *The first bin contains 3 items. Then all subsequent 1-bins with two items have smallest item $> 1 - 3x$ by diagram 8.35. So take $n_2 = 0$ for this case.*

2) *The first bin contains 2 items, with smallest item $> 1 - 3x$. Then all subsequent 1-bins with two items have smallest item $> 1 - 3x$ by diagram 8.34. So take $n_2 = 0$ for this case.*

3) *The first $n_2 \geqslant 1$ bins each contain 2 items with smallest item $\leqslant 1 - 3x$. Then, by diagram 8.34, these bins are identical. Bin $n_2 + 1$ contains (by assumption) either three items, or two items with smallest item $> 1 - 3x$, and thus all subsequent bins have smallest item $> 1 - 3x$.*

Since $z \leqslant 3x$ and $v \leqslant 1 - 3x$, it must be that v is the largest item $\leqslant 1 - 3x$ by the FFD-rule. □

8.9.1.3 Principal cases

The configuration in diagram 8.32 is analysed by conditioning on the size of the largest item z. This gives three principal cases.

1) $z \leqslant \frac{1}{2}$, this implies that block 1 and 2 are both empty.

2) $z > 3x$; block 1 is not empty.

3) $\frac{1}{2} < z \leqslant 3x$; block 1 is empty and block 2 is not.

The second and third case are broken down further. An overview of the bounds for these cases and subcases is given in table 8.14 (p. 132).

8.9.2 No 1-item[10]

We can derive a bound for this case under a slightly more general condition; we take $x \in \langle \frac{1}{6}, \frac{1}{5}]$ instead of $x \in \langle \frac{2}{11}, \frac{1}{5}]$. We have the following configuration.

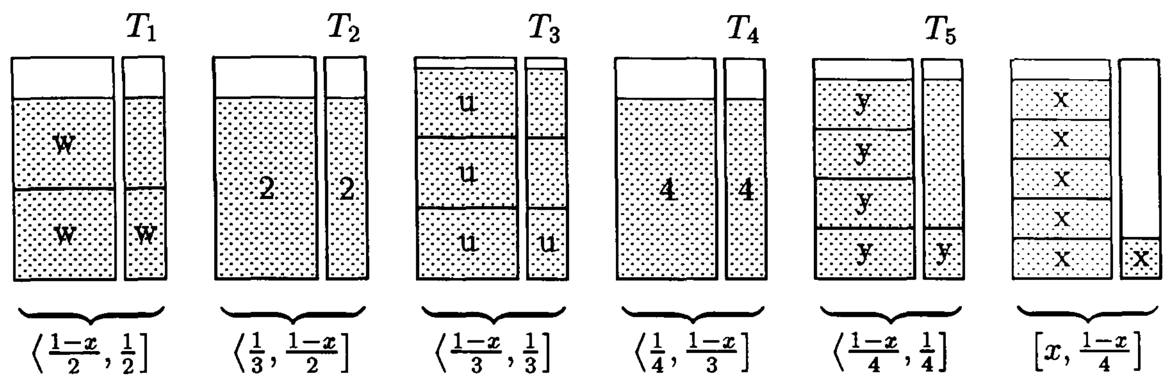


Diagram 8.37. General bin-configuration for list without 1-items and $x \in \langle \frac{1}{6}, \frac{1}{5}]$.

We use the generic weighting function with $W(x) = \frac{1}{5}$. The maximum pattern-weight (ratio) under this weighting function is $\frac{71}{60}$. The set-packing problem on the transition bins gives an upper bound for the constant of $\frac{77}{60}$. Combining this gives the following bound.

$$FFD \leqslant \tfrac{77}{60} + \tfrac{71}{60} CSP_R \tag{8.50}$$

The example in diagram 8.38 shows that this bound is asymptotically tight.

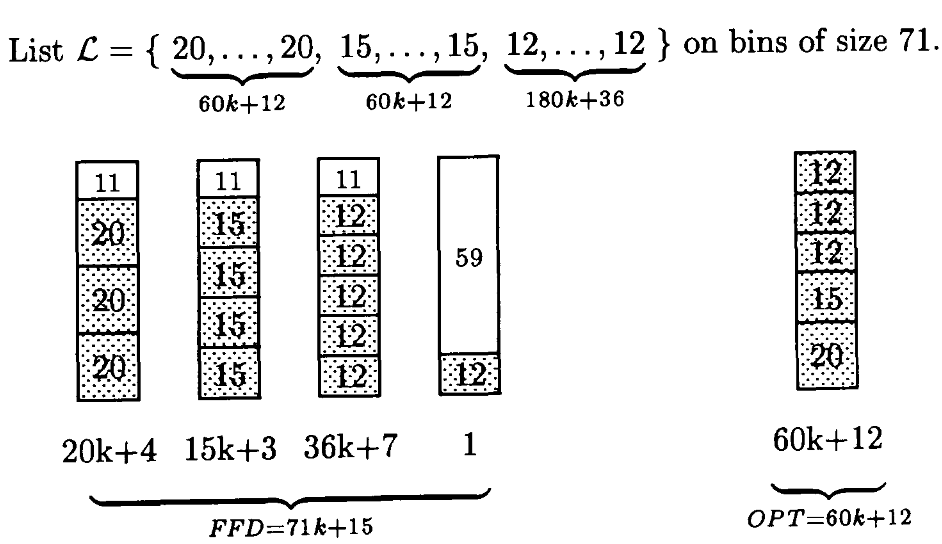


Diagram 8.38. Example for $FFD = \frac{4}{5} + \frac{71}{60} OPT$.

[10]Details of derivations can be found in appendix D.2.1 (p. 203)

8.9.3 $\exists$ 1-item $> 3x$

We have the following configuration. By the FFD-rule there are no items with size in $\langle v, 1-z]$

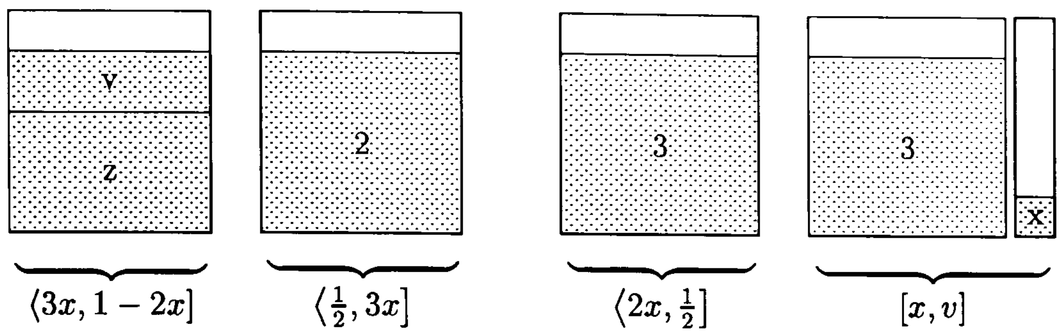


Diagram 8.39. Minimal bin-configuration for list with 1-item $> 3x$ and $x \in \langle \frac{2}{11}, \frac{1}{5} \rangle$.

in the list. As shown in section 8.9.1, item v is the largest item strictly less than $2x$. Together with $z \leqslant 1 - 2x$, this implies that there are certainly no items in the interval $\langle v, 2x]$ in the list. Furthermore $v > x$ must hold, otherwise FFD would have placed another item in the first bin. Note that $\frac{1}{3} < 2x \leqslant \frac{1-x}{2}$.

Ranges for v We analyse this configuration by conditioning on the size of v, which will give a further 5 cases. The range of $\langle x, 2x \rangle$ is split into 5 intervals as shown in table 8.12. This split was

case	v-range	$W(z)$
1	$v > \frac{1-x}{3}$	$\frac{2}{3}$
2	$v \in \langle \frac{1}{4}, \frac{1-x}{3}]$	$\frac{11}{16}$
3	$v \in \langle \frac{1-3x}{2}, \frac{1}{4}]$	$\frac{7}{10}$
4	$v \in \langle \frac{1-x}{4}, \frac{1-3x}{2}]$	$\frac{59}{80}$
5	$v \in \langle x, \frac{1-x}{4}]$	$\frac{31}{40}$

Table 8.12. Cases for 1-item $> 3x$
$W(z)$ is the weight of the (largest) 1-item

determined by a 'trial and error' process. One starts with a range for v and determines the bound that results from it. If the ratio (in this bound) is tight or the bound is sufficiently tight to prove bound (8.1) we stop. If this is not the case then we check what patterns achieve the ratio in the bound. This gives a requirement on v in terms of x, for each of these patterns to be valid. This in turn indicates a further subdivision of the interval (for an example see the 'valid for' column in table D.22 on page 213). This process ended with the division shown in table 8.12.

Weighting function For each case we use the weights $W(z) = V$ and $W(v) = 1 - V$ for the items in the first bin(s). This ensures that the bin weight of all bins in block 1 is exactly 1. The parameter V is determined separately for each case. For all other items we use the generic weighting function and the extension $W(s) = \frac{3}{5}$ (see page 206) to cover the 1-items in $\langle\frac{1}{2}, 3x]$. This ensures that all bins in block 2 have a bin weight of at least one (use lemma 8.28). By virtue of the generic weighting function all recurrent bins in block 3 have a bin weight of at least one.

For cases 2–5 we strengthen the weighting function to derive a tighter bound. In order to do so we develop a more suitable formulation of [the constraints in] the ratio problem. This is done in appendix D.2.2. This formulation, program (D51), is used as the basis for cases 2–5 to arrive at the stronger weighting functions, which are used in this section.

8.9.3.1 $z > 3x$ and $v > \frac{1-x}{3}$

We use weights $W(z) = \frac{2}{3}$ and $W(v) = \frac{1}{3}$, and the generic weighting function with $W_x = \frac{1}{5}$ for the other items. An upper bound for the ratio is given by the following program.

$$r \leqslant \begin{array}{|ll|}\hline \text{Max} & \frac{2}{3}a_1 + \frac{1}{3}a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{3}a_6 + \frac{4}{15}a_7 + \frac{1}{4}a_8 + \frac{1}{5}a_9 \\ \text{st} & za_1 + va_2 + \left(\frac{1}{2}\right)^+ a_3 + \left(\frac{1-x}{2}\right)^+ a_4 + \left(\frac{1}{3}\right)^+ a_5 + \left(\frac{1-x}{3}\right)^+ a_6 \\ & \qquad + \left(\frac{1}{4}\right)^+ a_7 + \left(\frac{1-x}{4}\right)^+ a_8 + xa_9 \leqslant 1 \\ & z > 3x \text{ and } v > \frac{1-x}{3} \\ & x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle \text{ and } a_i \in \mathbb{N} \\ \hline \end{array} \tag{8.51}$$

Substituting the lower bounds for z and v, and realising that at least one of $a_1, \ldots, a_8$ must be non-zero to achieve a value greater than 1, means that we can simplify the first constraint to

$$3xa_1 + \tfrac{1-x}{3}a_2 + \tfrac{1}{2}a_3 + \tfrac{1-x}{2}a_4 + \tfrac{1}{3}a_5 + \tfrac{1-x}{3}a_6 + \tfrac{1}{4}a_7 + \tfrac{1-x}{4}a_8 + xa_9 < 1 \tag{8.52}$$

Since for $x > \frac{2}{11}$ we have $3x > \frac{2}{3}(1-x)$, we may assume $a_1 = 0$ (dominance by a_6). And obviously we may assume $a_2 = 0$ (dominance by a_6). When $a_3 = 0$, the resulting program over 6 variables is exactly the same as for the case when there are no 1-items (see section 8.9.2). When $a_3 = 1$, the resulting program has a value of $\frac{71}{60}$ by section D.2.1 ('Extension of weighting function'). Ergo, $\frac{71}{60}$ is an upper bound for the ratio. Since we have used the same weights for the items in $\left[x, \frac{1}{2}\right]$ as for the case with no 1-items we have the same upper bound for the constant; viz. $\frac{77}{60}$. Combining this gives the following bound.

$$FFD \leqslant \tfrac{77}{60} + \tfrac{71}{60} CSP_R \tag{8.53}$$

8.9.3.2 $z > 3x$ and $v \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right]$

There are no items in $\langle v, 2x]$ in the list, so that there certainly no items in $\langle \frac{1-x}{3}, 2x]$ in the list. We use the following weighting function (see pages 210–211 for details).

$$W(z) = \tfrac{11}{16}, W(v) = \tfrac{5}{16} \text{ and } W(s) = \begin{cases} \frac{20}{32}, & s \in \left\langle \frac{1}{2}, 3x \right] \\ \frac{16}{32}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{14}{32}, & s \in \left\langle 2x, \frac{1-x}{2} \right] \\ \frac{9}{32}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{8}{32}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{7}{32}, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \tag{8.54}$$

The maximum pattern-weight under this weighting function is 19/16. An upper bound for the constant is determined as 1. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.40.

$$FFD \leqslant 1 + \tfrac{19}{16} CSP_R \tag{8.55}$$

8.9.3.3 $z > 3x$ and $v \in \left\langle \frac{1-3x}{2}, \frac{1}{4} \right]$

There are no items in $\langle v, 2x]$ in the list, so that there certainly no items in $\langle \frac{1}{4}, 2x]$ in the list. We use the following weighting function (see pages 212–213 for details).

$$W(z) = \tfrac{7}{10}, W(v) = \tfrac{3}{10} \text{ and } W(s) = \begin{cases} \frac{36}{60}, & s \in \left\langle \frac{1}{2}, 3x \right] \\ \frac{31}{60}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{26}{60}, & s \in \left\langle 2x, \frac{1-x}{2} \right] \\ \frac{15}{60}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{13}{60}, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \tag{8.56}$$

The maximum pattern-weight under this weighting function is 6/5. An upper bound for the constant is determined as 1. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.41.

$$FFD \leqslant 1 + \tfrac{6}{5} CSP_R \tag{8.57}$$

8.9.3.4 $z > 3x$ and $v \in \langle \frac{1-x}{4}, \frac{1-3x}{2}]$

There are no items in $\langle v, 2x]$ in the list, so that there certainly no items in $\langle \frac{1-3x}{2}, 2x]$ in the list. We use the following weighting function (see pages 214–215 for details).

$$W(z) = \tfrac{59}{80}, W(v) = \tfrac{21}{80} \text{ and } W(s) = \begin{cases} \frac{53}{80}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{42}{80}, & s \in \langle \frac{1-x}{2}, \frac{1}{2}] \\ \frac{32}{80}, & s \in \langle 2x, \frac{1-x}{2}] \\ \frac{20}{80}, & s \in \langle \frac{1-x}{4}, \frac{1-3x}{2}] \\ \frac{16}{80}, & s \in [x, \frac{1-x}{4}] \end{cases} \tag{8.58}$$

The maximum pattern-weight under this weighting function is 19/16. An upper bound for the constant is determined as 9/8. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.42.

$$FFD \leqslant \tfrac{9}{8} + \tfrac{19}{16} CSP_R \tag{8.59}$$

8.9.3.5 $z > 3x$ and $v \in \langle x, \frac{1-x}{4}]$

There are no items in $\langle v, 2x]$ in the list, so that there certainly no items in $\langle \frac{1-x}{4}, 2x]$ in the list. We use the following weighting function (see pages 216–217 for details).

$$W(z) = \tfrac{31}{40}, W(v) = \tfrac{9}{40} \text{ and } W(s) = \begin{cases} \frac{27}{40}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{20}{40}, & s \in \langle \frac{1-x}{2}, \frac{1}{2}] \\ \frac{18}{40}, & s \in \langle 2x, \frac{1-x}{2}] \\ \frac{8}{40}, & s \in [x, \frac{1-x}{4}] \end{cases} \tag{8.60}$$

The maximum pattern-weight under this weighting function is 47/40. An upper bound for the constant is determined as 9/10. This gives the following bound, which is asymptotically tight as shown by the instance in diagram 8.43.

$$FFD \leqslant \tfrac{9}{10} + \tfrac{47}{40} CSP_R \tag{8.61}$$

List $\mathcal{L} = \{\ \underbrace{22,\ldots,22}_{12k+9},\ \underbrace{10,\ldots,10}_{12k+9},\ \underbrace{8,\ldots,8}_{28k+20},\ 7\ \}$ on bins of size 38.

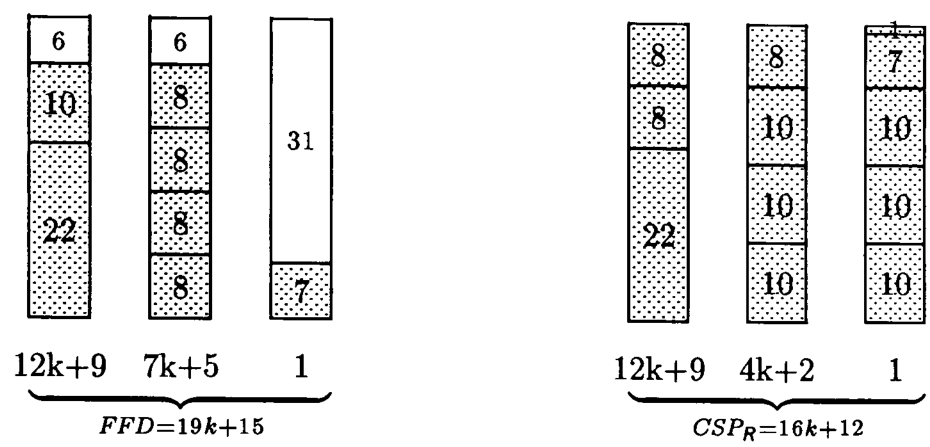


Diagram 8.40. $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle$, $z > 3x$ and $v \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right]$; instance with $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$.

List $\mathcal{L} = \{\ \underbrace{28,\ldots,28}_{4k+1},\ \underbrace{12,\ldots,12}_{4k+1},\ \underbrace{10,\ldots,10}_{8k+4},\ 9\ \}$ on bins of size 48.

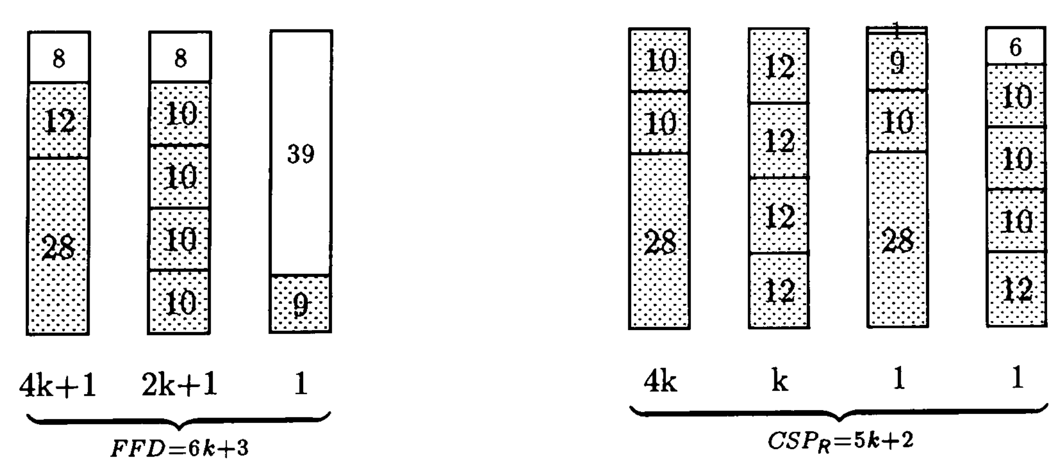


Diagram 8.41. $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle$, $z > 3x$ and $v \in \left\langle \frac{1-3x}{2}, \frac{1}{4} \right]$; instance with $FFD = \frac{3}{5} + \frac{6}{5} CSP_R$.

List $\mathcal{L} = \{\ \underbrace{104,\ldots,104}_{12k+9},\ \underbrace{36,\ldots,36}_{12k+9},\ \underbrace{35,\ldots,35}_{12k+8},\ \underbrace{31,\ldots,31}_{20k+16}\ \}$ on bins of size 170.

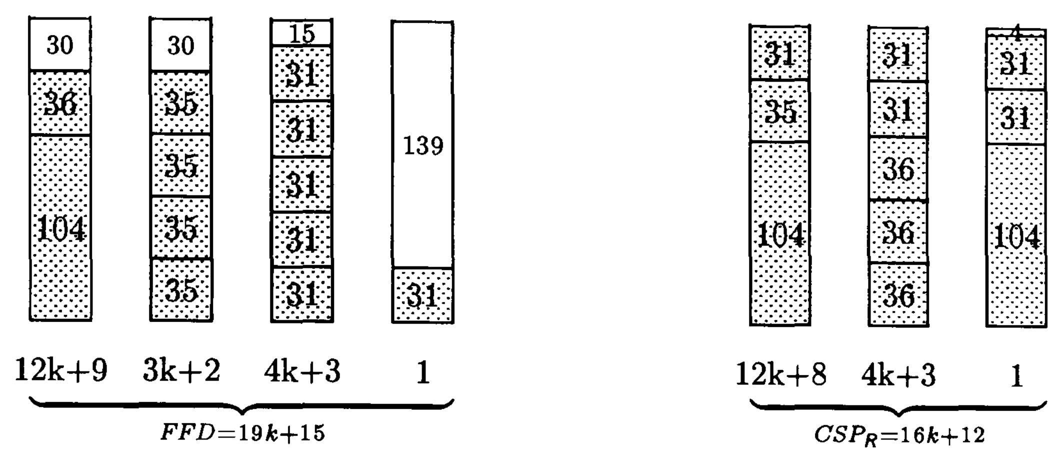


Diagram 8.42. $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle$, $z > 3x$ and $v \in \left\langle \frac{1-x}{4}, \frac{1-3x}{2} \right]$; instance with $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$.

List $\mathcal{L} = \{\ \underbrace{52,\ldots,52}_{30k+18},\ \underbrace{34,\ldots,34}_{10k+6},\ \underbrace{16,\ldots,16}_{30k+18},\ \underbrace{15,\ldots,15}_{60k+36}\ \}$ on bins of size 82.

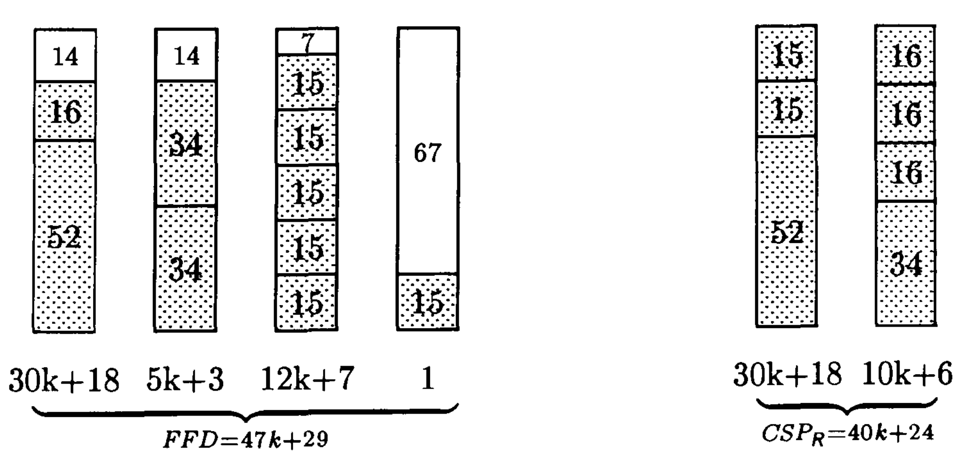
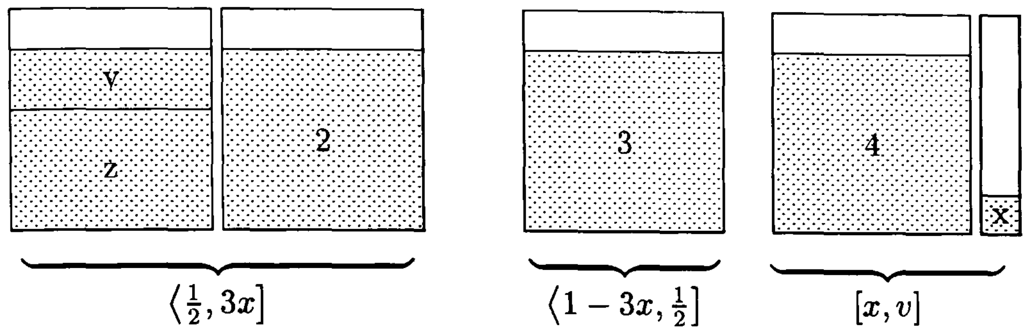


Diagram 8.43. $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle$, $z > 3x$ and $v \in \left\langle x, \frac{1-x}{4} \right]$; instance with $FFD = \frac{4}{5} + \frac{47}{40} CSP_R$.

8.9.4 All 1-items $\leqslant 3x$

We have the following configuration.



By the FFD-rule there are no items with size in $\langle v, 1-z]$ in the list. As shown in section 8.9.1, item v is the largest item less than or equal to $1-3x$. Together with $z \leqslant 3x$, this implies that there are certainly no items in the interval $\langle v, 1-3x]$ in the list. Furthermore $v > 1-4x$ must hold for the first bin to have two items. Note that $\frac{1-x}{2} < 1-3x < \frac{1}{2}$.

Ranges for v We analyse this case by conditioning on the size of v, which will give a further 5 cases. The range of $\langle 1-4x, 1-3x]$ is split into 5 intervals as shown in table 8.13. As in the previous

case	v-range
1	$v > \frac{1}{3}$, *or* no item v
2	$v \in \langle \frac{1+x}{4}, \frac{1}{3}]$
3	$v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$
4	$v \in \langle \frac{1-x}{3}, \frac{7-19x}{12}]$
5	$v \in \langle 1-4x, \frac{1-x}{3}]$

Table 8.13. Cases for 1-item $\leqslant 3x$

section this division was determined by a 'trial and error' process.

Weighting Function We use the same weights and weighting function as in section 8.9.3. By diagram 8.36 we have that bins in block 2 contain either two or three items. In the former case the smallest item is $> 1-3x$, so all bins in this block have a bin weight of at least one. As in the previous section all other bins have bin weight of at least one.

8.9.4.1 $z \leqslant 3x$ and $\{v > \frac{1}{3}$ or no item $v\}$

We use weights $W(z) = \frac{3}{5}$ and $W(v) = \frac{2}{5}$, and the generic weighting function with $W(x) = \frac{1}{5}$ for the other items. An upper bound for the ratio is given by the following program.

$$r \leqslant \boxed{\begin{array}{ll} \text{Max} & \frac{3}{5}a_1 + \frac{2}{5}a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{3}a_6 + \frac{4}{15}a_7 + \frac{1}{4}a_8 + \frac{1}{5}a_9 \\ \text{st} & za_1 + va_2 + \left(\frac{1}{2}\right)^+ a_3 + \left(\frac{1-x}{2}\right)^+ a_4 + \left(\frac{1}{3}\right)^+ a_5 + \left(\frac{1-x}{3}\right)^+ a_6 \\ & \qquad + \left(\frac{1}{4}\right)^+ a_7 + \left(\frac{1-x}{4}\right)^+ a_8 + xa_9 \leqslant 1 \\ & z > \frac{1}{2} \text{ and } v > \frac{1}{3} \\ & x \in \langle \frac{2}{11}, \frac{1}{5}] \text{ and } a_i \in \mathrm{N} \end{array}} \qquad (8.62)$$

We may assume $a_1 = a_2 = 0$, because of dominance by a_3 and a_5 respectively. When there is no item v (and no item z) one already has $a_1 = a_2 = 0$. In both cases the resulting program is the same as the one resulting from (8.51). Since we have used the same weights for the items in $[x, \frac{1}{2}]$ as for the case when there are no 1-items we have the same upper bound for the constant. This gives the following bound for both cases.

$$FFD \leqslant \frac{77}{60} + \frac{71}{60} CSP_R \qquad (8.63)$$

8.9.4.2 $z \leqslant 3x$ and $v \in \langle \frac{1+x}{4}, \frac{1}{3}]$

There are no items in $\langle v, 1 - 3x]$ in the list, so that there certainly no items in $\langle \frac{1}{3}, 1 - 3x]$ in the list. We use the following weighting function (see pages 220–221 for details).

$$W(s) = \begin{cases} \frac{15}{24}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{12}{24}, & s \in \langle 1 - 3x, \frac{1}{2}] \\ \frac{9}{24}, & s \in \langle \frac{1+x}{4}, \frac{1}{3}] \\ \frac{8}{24}, & s \in \langle \frac{1-x}{3}, \frac{1+x}{4}] \\ \frac{7}{24}, & s \in \langle \frac{1}{4}, \frac{1-x}{3}] \\ \frac{6}{24}, & s \in \langle \frac{1-x}{4}, \frac{1}{4}] \\ \frac{5}{24}, & s \in [x, \frac{1-x}{4}] \end{cases} \qquad (8.64)$$

The maximum pattern-weight under this weighting function is 29/24. An upper bound for the constant is determined as 7/6. This gives the bound $FFD \leqslant \frac{7}{6} + \frac{29}{24} CSP_R$. Although this bound is asymptotically tight as shown by the instance in diagram 8.45, it is not sufficient to prove bound (8.1). A slight refinement of the weighting function, based upon the first transition-bin allows a sharpening of the constant to give the following bound.

$$FFD \leqslant \frac{13}{12} + \frac{29}{24} CSP_R \qquad (8.65)$$

8.9.4.3 $z \leqslant 3x$ and $v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$

Observe that $x > \frac{2}{11}$ is a necessary condition for the interval to be non-void. Item v is the largest item less than or equal to $1 - 3x$. There are no items in the interval $\langle v, 1-3x]$, so that there certainly are no items in $\langle \frac{1+x}{4}, 1-3x]$. This gives the following configuration.

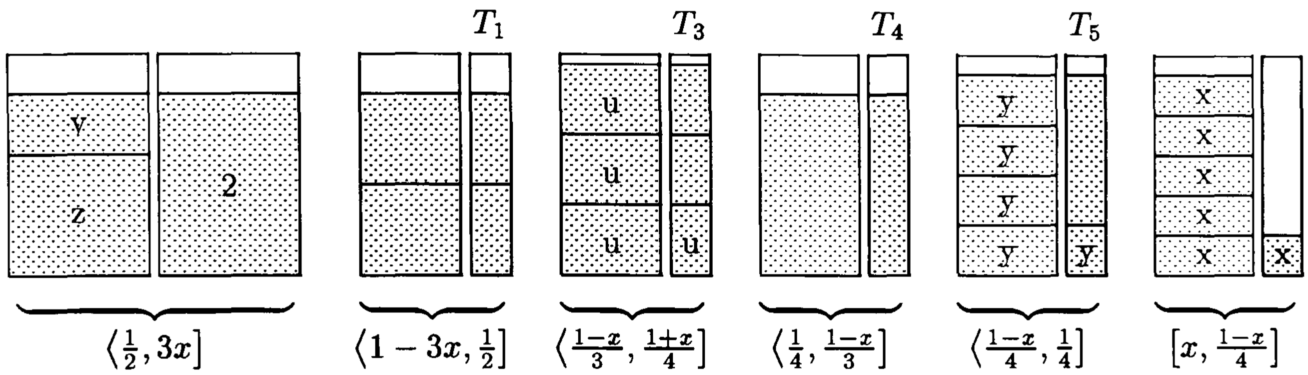


Diagram 8.44. Minimal bin configuration for $\frac{11}{9}$-ratio

Unfortunately, any further partitioning of the range of v in terms of x does not directly enable us to reduce the ratio to $\frac{11}{9}$ or less. We analyse this case by conditioning on the existence of bins with largest item $y \in \langle \frac{1-x}{4}, \frac{1}{4}]$ to give two cases. We show that the following bound holds.

$$FFD \leqslant \tfrac{35}{36} + \tfrac{11}{9} CSP_R. \tag{8.66}$$

8.9.4.3a There is no bin with largest item in $\langle \frac{1-x}{4}, \frac{1}{4}]$ We use the following weights and weighting function (see pages 223–224 for details).

$$W(z) = \tfrac{59}{90}, W(v) = \tfrac{31}{90} \text{ and } W(s) = \begin{cases} \frac{3}{5}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{1}{2}, & s \in \langle 1-3x, \frac{1}{2}] \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1+x}{4}] \\ \frac{4}{15}, & s \in \langle \frac{1}{4}, \frac{1-x}{3}] \\ \frac{1}{5}, & s \in [x, \frac{1}{4}] \end{cases} \tag{8.67}$$

The maximum pattern-weight under this weighting function is 107/90. An upper bound for the constant is determined as $\frac{16}{15}$. This gives the following bound.

$$FFD \leqslant \tfrac{16}{15} + \tfrac{107}{90} CSP_R. \tag{8.68}$$

That bound (8.66) holds for this case follows from a direct comparison for $CSP_R \geqslant 2\frac{5}{6}$ and corollary 8.20 otherwise. Note that this bound rules out the cases $(FFD, OPT) = (18+11k, 14+9k)$ and $(13+11k, 10+9k)$ (see table 8.2), and thus that $FFD \leqslant 2/3 + (11/9)OPT$ holds for this subcase.

8.9.4.3b There is a bin with largest item in $\langle\frac{1-x}{4},\frac{1}{4}]$ By assumption 8.16 we have that there are no bins with largest item in the interval $\langle\frac{1}{4},\frac{1-y}{3}]$. This implies that we can replace the range $\langle\frac{1}{4},\frac{1-x}{3}]$ by $\langle\frac{1-y}{3},\frac{1-x}{3}]$ in diagram 8.44. We use the following weights and weighting function (see pages 225–228 for details).

$$W(z)=\tfrac{23}{36}, W(v)=\tfrac{13}{36} \text{ and } W(s)=\begin{cases}\frac{11}{18}, & s\in\langle\frac{1}{2},3x]\\ \frac{1}{2}, & s\in\langle 1-3x,\frac{1}{2}]\\ \frac{1}{3}, & s\in\langle\frac{1-x}{3},\frac{1+x}{4}]\\ \frac{19}{72}, & s\in\langle\frac{1-y}{3},\frac{1-x}{3}]\\ \frac{1}{4}, & s\in[y,\frac{1-y}{3}]\\ \frac{17}{72}, & s\in[x,y\rangle\end{cases} \tag{8.69}$$

The maximum pattern-weight under this weighting function is 11/9. An upper bound for the constant is determined as 35/36. This gives the following bound, which is asymptotically tight as shown by the instances in diagram 8.46. We note that, under weighting function (8.69), these instances also satisfy $FFD(\mathcal{L})=35/36+W(\mathcal{L})$.

$$FFD\leqslant\tfrac{35}{36}+\tfrac{11}{9}CSP_R. \tag{8.70}$$

8.9.4.4 $z \leqslant 3x$ and $v \in \left\langle \frac{1-x}{3}, \frac{7-19x}{12} \right]$

There are no items in $\langle v, 1-3x]$ in the list, so that there certainly are no items in $\left\langle \frac{7-19x}{12}, 1-3x \right]$ in the list. We use the following weighting function (see pages 229–230 for details).

$$W(z) = \tfrac{119}{180}, W(v) = \tfrac{61}{180} \text{ and } W(s) = \begin{cases} \frac{3}{5}, & s \in \left\langle \frac{1}{2}, 3x \right] \\ \frac{1}{2}, & s \in \left\langle 1-3x, \frac{1}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-x}{3}, \frac{7-19x}{12} \right] \\ \frac{4}{15}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{1}{5}, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \tag{8.71}$$

The maximum pattern-weight under this weighting function is 43/36. An upper bound for the constant is determined as 67/60. This gives the following bound.

$$FFD \leqslant \tfrac{67}{60} + \tfrac{43}{36} CSP_R \tag{8.72}$$

8.9.4.5 $z \leqslant 3x$ and $v \in \left\langle 1-4x, \frac{1-x}{3} \right]$

Note that $x > \frac{2}{11}$ must hold for this interval to be non-void. There are no items in $\langle v, 1-3x]$ in the list, so that there certainly are no items in $\left\langle \frac{1-x}{3}, 1-3x \right]$ in the list. We use the following weighting function (see pages 231–232 for details).

$$W(z) = \tfrac{209}{300}, W(v) = \tfrac{91}{300} \text{ and } W(s) = \begin{cases} \frac{364}{600}, & s \in \left\langle \frac{1}{2}, 3x \right] \\ \frac{364}{600}, & s \in \left\langle 1-3x, \frac{1}{2} \right] \\ \frac{160}{600}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{150}{600}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{139}{600}, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \tag{8.73}$$

The maximum pattern-weight under this weighting function is 91/75. An upper bound for the constant is determined as 33/40. This gives the following bound.

$$\text{all 1-items} \leqslant 3x,\ v \in \left\langle 1-4x, \tfrac{1-x}{3} \right] \ \Rightarrow\ FFD \leqslant \tfrac{33}{40} + \tfrac{91}{75} CSP_R \tag{8.74}$$

List $\mathcal{L} = \{ \underbrace{38, \ldots, 38}_{12k+3}, \underbrace{23, \ldots, 23}_{12k+3}, \underbrace{20, \ldots, 20}_{24k+6}, \underbrace{15, \ldots, 15}_{36k+8}, 14 \}$ on bins of size 73.

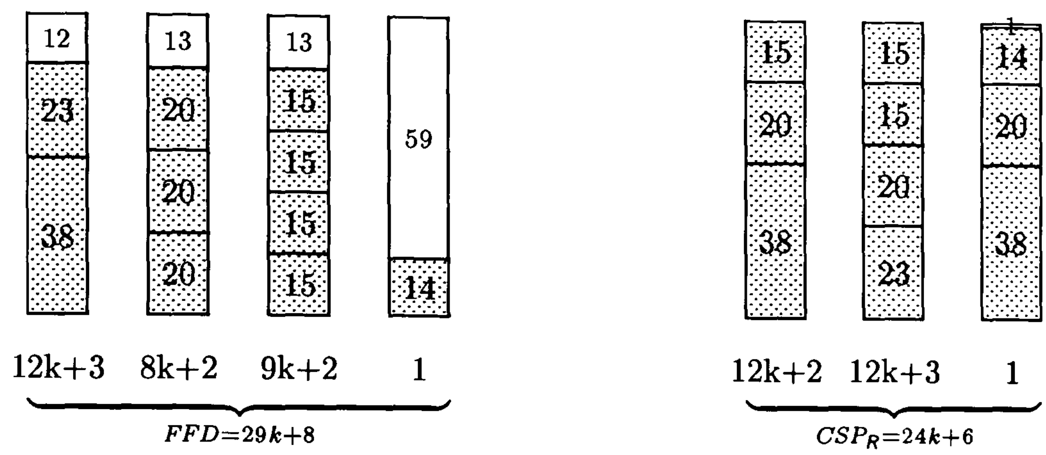


Diagram 8.45. $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z \leqslant 3x$ and $v \in \langle \frac{1+x}{4}, \frac{1}{3}]$; instance with $FFD = \frac{3}{4} + \frac{29}{24} CSP_R$.

List $\mathcal{L} = \{ \underbrace{72, \ldots, 72}_{6k+10}, \underbrace{41, \ldots, 41}_{6k+10}, \underbrace{39, \ldots, 39}_{6k+7}, 37, 37, \underbrace{29, \ldots, 29}_{12k+17}, \underbrace{28, \ldots, 28}_{4} \}$ on bins of size 140.

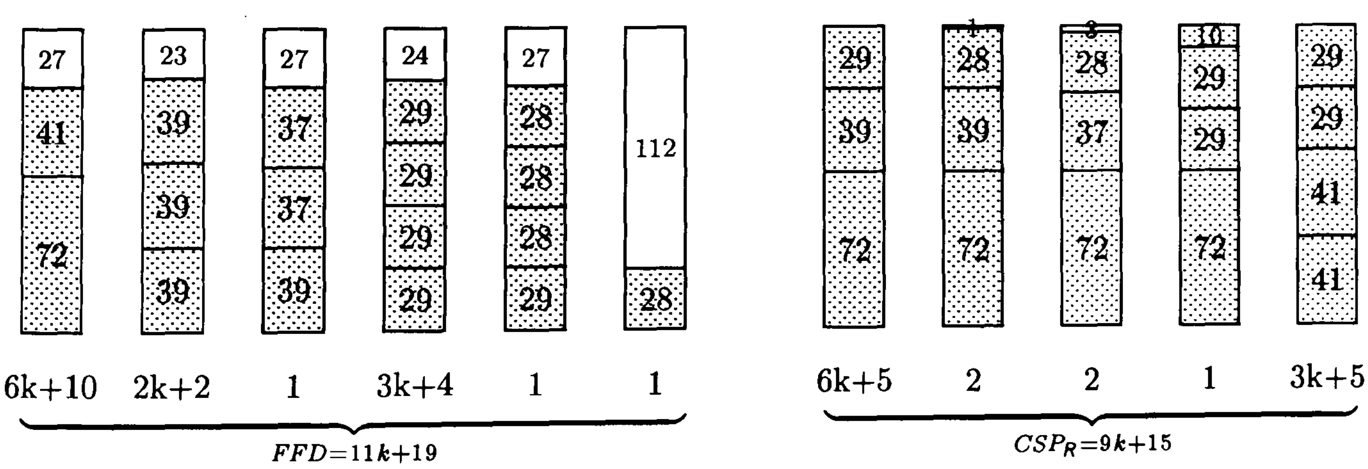


Diagram 8.46. $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z \leqslant 3x$ and $v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$; instance with $FFD = \frac{2}{3} + \frac{11}{9} CSP_R$.

8.9.5 Overview of bounds for $x \in \langle \frac{2}{11}, \frac{1}{5}]$

	Case		Bound	r	$\frac{8}{9}$	$\frac{7}{9}$	
1	no 1-item		$FFD \leqslant \frac{77}{60}+\frac{71}{60}CSP_R$		—	(13,10)	√
2	∃ 1-item > 3x	$v > \frac{1-x}{3}$	$FFD \leqslant \frac{77}{60}+\frac{71}{60}CSP_R$		—	(13,10)	√
3		$v \in \langle \frac{1}{4}, \frac{1-x}{3}]$	$FFD \leqslant 1+\frac{19}{16}CSP_R$	*	—	—	√
4		$v \in \langle \frac{1-3x}{2}, \frac{1}{4}]$	$FFD \leqslant 1+\frac{6}{5}CSP_R$	*	—	(13,10)	√
5		$v \in \langle \frac{1-x}{4}, \frac{1-3x}{2}]$	$FFD \leqslant \frac{9}{8}+\frac{19}{16}CSP_R$	*	—	(13,10)	√
6		$v \in \langle x, \frac{1-x}{4}]$	$FFD \leqslant \frac{9}{10}+\frac{47}{40}CSP_R$	*	—	—	√
7	∀ 1-item ⩽ 3x	$v > \frac{1}{3}$, no item v	$FFD \leqslant \frac{77}{60}+\frac{71}{60}CSP_R$		—	(13,10)	√
8		$v \in \langle \frac{1+x}{4}, \frac{1}{3}]$	$FFD \leqslant \frac{13}{12}+\frac{29}{24}CSP_R$	*	(18,14)	(24,19)(13,10)	√
9		$v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$	$FFD \leqslant \frac{35}{36}+\frac{11}{9}CSP_R$	*	not applicable		√
10		$v \in \langle \frac{1-x}{3}, \frac{7-19x}{12}]$	$FFD \leqslant \frac{67}{60}+\frac{43}{36}CSP_R$		—	(13,10)	√
11		$v \in \langle 1-4x, \frac{1-x}{3}]$	$FFD \leqslant \frac{33}{40}+\frac{91}{75}CSP_R$		—	—	√

Table 8.14. FFD-bounds for $\beta(x) = 5$
A tick indicates that for this case bound (8.1) holds.

The bounds for the various subcases are summarised in the above table. An asterisk in the 'ratio' column indicates that this bound is proven to be asymptotically sharp. The corresponding examples can be found in the appendix.

Our primary aim is to prove that bound (8.1) holds. This bound follows from the slightly tighter bound (8.75). To prove the latter we use corollary 8.20, which implies that we only need to check for $CSP_R > 8\frac{1}{5}$. A straightforward comparison for all cases will now show that the bound also holds. This proves, taking into account assumption 8.22, the following bound.

$$x \in \langle \tfrac{2}{11}, \tfrac{1}{5}] \Rightarrow FFD \leqslant \tfrac{35}{36} + \tfrac{11}{9} CSP_R \tag{8.75}$$

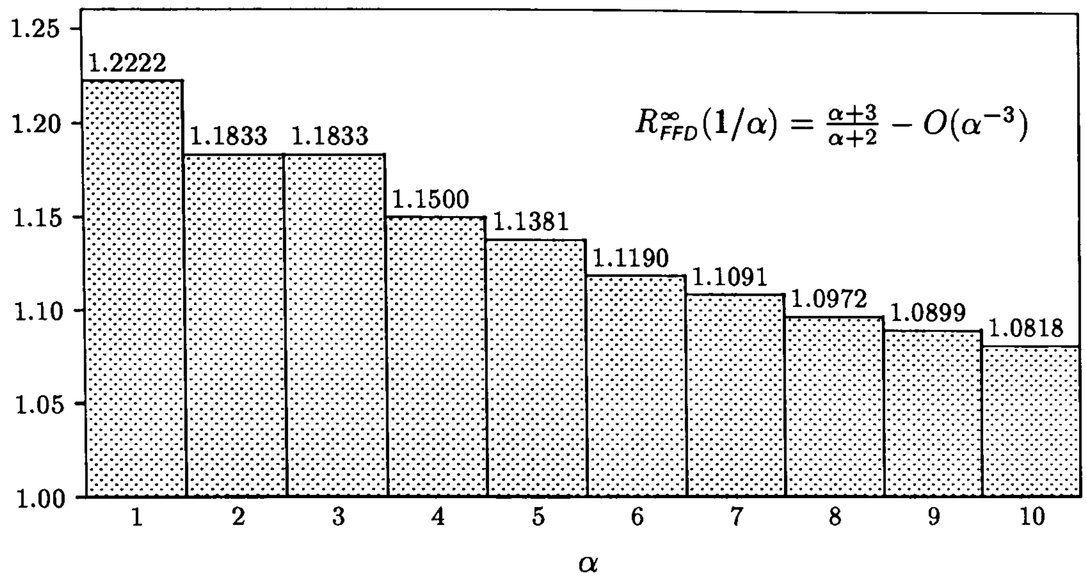


Diagram 8.47. FFD-ratio for $\mathcal{L} \subset \langle 0, \frac{1}{\alpha}]$.

8.10 Lists with no 1-items

In this section we will derive bounds for FFD when the largest item in the list is at most half the bin size. Throughout we assume that we have a list with largest item $\varphi \in \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ and that FFD packs this list into a configuration with critical item $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$ (note that $2 \leqslant \alpha \leqslant \beta$ and $\alpha, \beta \in \mathbb{N}$).

8.10.1 Known results

Johnson [39, p302] gives the following bounds for the parametric asymptotic ratio of FFD.

$$F_{\alpha} \equiv 1 + \frac{1}{\alpha+2} - \frac{2}{\alpha(\alpha+1)(\alpha+2)} \leqslant R^{\infty}_{FFD}(\varphi) \leqslant 1 + \frac{1}{\alpha+2} \tag{8.76}$$

He also[38] derived the following result

$$R^{\infty}_{FFD}(\varphi) = \begin{cases} \frac{71}{60}, & \frac{8}{29} < \varphi \leqslant \frac{1}{2} \\ \frac{7}{6}, & \frac{1}{4} < \varphi \leqslant \frac{8}{29} \\ \frac{23}{20}, & \frac{1}{5} < \varphi \leqslant \frac{1}{4} \end{cases} \tag{8.77}$$

and conjectured that for $\varphi = 1/\alpha$ and integers $\alpha \geqslant 4$ the lower bound in (8.76) is tight.

Csirik[17] showed this to be true when α is even, but false when α is odd. For the latter case he showed that the ratio is given by $G_{\alpha} \equiv 1 + \frac{1}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}$. His proof is a continuation of the methods used in Johnson,[38] and Johnson et al.[37] He uses a weighting function in terms of individual items, defines different 'discounts' for the weight of a bin, depending upon the combination of items it contains, and finally uses a weighting function for an entire list, defined in terms of how this list can be partioned into one- and two-element sets. The majority of his paper is devoted to proving (through an extensive case-analysis) a lemma, which in effect gives an upper bound for the maximum bin-weight. Combined with a generalisation of his Lemma 4.2.[37] he proves his main theorem:

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha}\right] \text{ and } \alpha \geqslant 5 \quad \Rightarrow \quad FFD(\mathcal{L}) \leqslant \begin{cases} 4 + F_\alpha OPT(\mathcal{L}), & \text{if } \alpha \text{ is even} \\ 4 + G_\alpha OPT(\mathcal{L}), & \text{if } \alpha \text{ is odd} \end{cases} \tag{8.78}$$

and gives instances[11] to show that these bounds are asymptotically tight. This and (8.77) prove his final theorem

$$\alpha \geqslant 3 \quad \Rightarrow \quad R^{\infty}_{FFD}(1/\alpha) = \begin{cases} F_\alpha, & \text{if } \alpha \text{ is even} \\ G_\alpha, & \text{if } \alpha \text{ is odd} \end{cases} \tag{8.79}$$

---

[11] These instances correspond to the 'homogeneous' solution in tables B.6 and B.7

[12] It was subsequently discovered[14, 40] that the general structure of the parametric asymptotic ratio of FFD has been determined by Xu[70] as follows, where d_α is defined as $\frac{(\alpha+1)^2}{\alpha^3+3\alpha^2+\alpha+1}$.

$$\varphi \in \left\langle \tfrac{1}{\alpha+1}, \tfrac{1}{\alpha}\right] \text{ and } \alpha \geqslant 3 \quad \Rightarrow \quad R^{\infty}_{FFD}(\varphi) = \begin{cases} F_\alpha, & \text{if } \frac{1}{\alpha+1} < \varphi \leqslant \frac{1}{\alpha} \text{ and } \alpha \text{ is even} \\ F_\alpha, & \text{if } \frac{1}{\alpha+1} < \varphi \leqslant d_\alpha \text{ and } \alpha \text{ is odd} \\ G_\alpha, & \text{if } d_\alpha < \varphi \leqslant \frac{1}{\alpha} \text{ and } \alpha \text{ is odd} \end{cases}$$

We do not know, what the constant is in her worst-case bounds.

8.10.2 New results

Although we sharpen the existing bounds, (8.78) the results in this section are not fundamentally new (the manner by which they are derived, however is). The purpose is to show that the asymptotic ratios of FFD are closely linked to a subset-sum problem on unit fractions, studied in section B.5, and to show that these ratios can be derived in a more elegant manner than hitherto known. To be more precise we show that $R^{\infty}_{FFD}(1/\alpha) = \max_{\beta \geqslant \alpha} S_\alpha(\beta)$, for $\alpha \geqslant 2$ and $S_\alpha(\beta)$ as defined in (B24). This leads to a more natural formulation [than (8.79)] of the asymptotic ratio.

8.30 **Corollary** ***(FFD-ratio)*** $\alpha \geqslant 3 \Rightarrow R^{\infty}_{FFD}(1/\alpha) = \frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$

The instances in tables B.6 and B.7 show that these asymptotic ratios are achievable. To visualise the behaviour of the parametric ratio, it is graphed in diagram 8.47.

Bounds In the remainder of this section we shall prove the following bounds:

$$\mathcal{L} \subset \langle 0, \tfrac{1}{2}] \Rightarrow FFD \leqslant \tfrac{9}{5} + \tfrac{71}{60} CSP_R \tag{8.80}$$

and

$$\mathcal{L} \subset \langle 0, \tfrac{1}{\alpha}] \text{ and } \alpha \geqslant 3 \Rightarrow FFD \leqslant \tfrac{\alpha+7}{\alpha+3} + \left(\tfrac{\alpha+3}{\alpha+2} - \tfrac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R \tag{8.81}$$

Preliminaries The bounds are derived using the generic weighting function with $W_x = \frac{1}{\beta}$. Note that this is a recurrent weighting function for $\beta \leqslant 2\alpha + 1$ (see the comment on page 90).

$$W(s) = \begin{cases} \frac{1}{i}, & s \in \langle \frac{1-x}{i}, \frac{1}{i}] \text{ and } \alpha \leqslant i \leqslant \beta - 1 \\ \frac{\beta-1}{\beta}\frac{1}{i}, & s \in \langle \frac{1}{i+1}, \frac{1-x}{i}] \text{ and } \alpha \leqslant i \leqslant \beta - 2 \\ \frac{1}{\beta}, & s \in [x, \frac{1-x}{\beta-1}] \end{cases} \tag{8.82}$$

A bound for the maximum pattern-weight [ratio] under this weighting function is given by the value of program (B12) (p. 167), which is analysed in section B.4

Constant A bound for the weight of a transition bin, under weighting function (8.82), is easily calculated. First determine a bound for the relative profit of an item as follows.

$$\rho(s) = W(s)/s \geqslant \min\{1, \tfrac{\beta-1}{\beta}\tfrac{1}{1-x}\} = \tfrac{\beta-1}{\beta}\tfrac{1}{1-x} \tag{8.83}$$

Since all [transition] bins are filled to a level $> 1 - x$, we get a lower bound for the bin weight as $W_{\text{bin}} > \min_s \rho(s) \times (1-x) = 1 - \frac{1}{\beta}$. There are at most $[(\beta-1)-(\alpha-1)] + [(\beta-2)-(\alpha-1)] = 2\beta - 2\alpha - 1$ transition bins, and this gives the following bound for the constant.

$$c < (2\beta - 2\alpha - 1) \times (1 - (1 - \tfrac{1}{\beta})) + (1 - W(x)) = 3 - 2\tfrac{\alpha+1}{\beta}. \tag{8.84}$$

Case	Bound	
$\beta = 2$	$FFD \leqslant \frac{1}{2} + CSP_R$	lemma 8.3 (page 81)
$\beta = 3$	$FFD \leqslant \frac{7}{9} + \frac{7}{6}\, CSP_R$	bound (8.24) (page 93)
$\beta = 4$	$FFD \leqslant \frac{13}{12} + \frac{7}{6}\, CSP_R$	bound (8.42) (page 108)
$\beta = 5$	$FFD \leqslant \frac{77}{60} + \frac{71}{60} CSP_R$	bound (8.50) (page 119)
$\beta = 6$	$FFD \leqslant \frac{9}{5} + \frac{71}{60} CSP_R$	bound (8.85) (page 136)
$\beta \geqslant 7$	$FFD < \frac{5}{6} + \frac{7}{6}\, Mat$	bound (8.14) (page 87)

Table 8.15. FFD-bounds for lists with largest item $\varphi \leqslant 1/2$ and critical item $x \in \langle \frac{1}{\beta+1}, \frac{1}{\beta}]$

8.10.3 $\alpha = 2$

To prove (8.80) we only need to investigate the case for $\beta = 6$, all the other cases are already covered in previous sections (see the summary in table 8.15).

8.10.3.1 $\beta = 6$

Since $\beta \leqslant 2\alpha + 1$ does not hold for this case, we cannot [directly] use the generic weighting function. This is because there can be 2-complete bins that have two additional items (see table 8.3, p. 90). However, these bins *cannot* occur in a *minimal* configuration. Such a bin contains two items in $\langle \frac{1}{3}, \frac{1}{2}]$ and two additional items $\geqslant x$. But then we can apply the cutting principle (assumption 8.13) to reduce this bin to a bin with 6 items of size x. This means that (8.82) is a recurrent weighting function for a [minimal] configuration. The maximum pattern-weight under this weighting function is given by the value of (B12), which is 6/5 (use lemma B22). A strict bound for the constant follows from (8.84) as 2. This proves the bound $FFD < 2 + \frac{6}{5} CSP_R$. However, it is known[37] that the asymptotic ratio for lists with no 1-items is $\frac{71}{60}$, so that this bound cannot be tight.

By conditioning on the existence of a bin with largest item $y \in \langle \frac{1-x}{5}, \frac{1}{5}]$ we will derive the following bound.

$$FFD \leqslant 1.8 + \tfrac{71}{60} CSP_R \tag{8.85}$$

8.10.3.1a There is a bin with largest item $y \in \langle \frac{1-x}{5}, \frac{1}{5}]$ It turns out that, rather than splitting the range $[x, \frac{1}{2}]$ in terms of x, it is necessary to split it according to y. This leads us to using a weighting function in terms of y; (D95). This is a slight variation of the generic weighting function, and can be found in appendix D.3.1 (p. 235). The maximum pattern-weight under this weighting function, is determined as 71/60.

To determine a bound for the constant, note that the relative profit of an item in $[y, 1/2]$ is at least $\frac{4}{5}\frac{1}{1-y}$. All the bins before [the bin with item] y are filled to a level strictly greater than $1-y$, and contain only items in $[y, 1/2]$. The weight of these transition bins is therefore $W_T > \frac{4}{5}$, which implies $W_T \geqslant \frac{49}{60}$ since 60 is a scalar for the weighting function. But there is no combination of items in $[y, 1/2]$ that has a weight of 49/60, so that $W_T \geqslant \frac{5}{6}$ must hold. There are at most 5 transition bins before [the bin with item] y. The weight of the transition bin with the item y is at least $\frac{1}{5} + 4 \times \frac{1}{6} = \frac{13}{15}$. Combining this gives the following bound for the constant; $c \leqslant 5 \times (1 - \frac{5}{6}) + (1 - \frac{13}{15}) + (1 - \frac{1}{6}) = 1.8$, and thus the bound $FFD \leqslant 1.8 + \frac{71}{60} CSP_R$.

8.10.3.1b There is no bin with largest item in $\langle \frac{1-x}{5}, \frac{1}{5}]$ We now split the range $[x, 1/2]$ according to x and use (D97) (p. 235) as a weighting function. The maximum pattern-weight, under this weighting function, is determined as 7/6.

The relative profit of items is at least 5/6, which gives a lower bound for the weight of the transition bins as $W_T > \frac{5}{6}(1-x) \geqslant \frac{25}{36}$, which implies $W_T \geqslant \frac{26}{36}$ since 36 is a scalar for the weighting function. There are at most 6 transition bins, which gives the following bound for the constant; $c \leqslant 6 \times (1 - \frac{26}{36}) + (1 - \frac{1}{6}) = 2.5$, and thus the bound $FFD < 2.5 + \frac{7}{6} CSP_R$.

Direct comparison will show that for $CSP_R \geqslant 42$ this bound implies bound (8.85). The naïve bound for this case gives $FFD < \frac{4}{5} + \frac{6}{5} CSP_R$, which implies bound (8.85) for $CSP_R \leqslant 60$. Ergo (8.85) holds for this case.

8.10.4 $\alpha \geqslant 3$

To prove (8.81) we only need to consider the cases for $\beta \in \{\alpha+1, \alpha+2, \alpha+3\}$, all the other cases are already covered in previous sections (see the summary in table 8.16).

8.10.4.1 $\beta = \alpha + 1$

We can use lemma B23, which shows that $\frac{\alpha+3}{\alpha+2} - \frac{2}{\alpha(\alpha+1)(\alpha+2)}$ is an upper bound for the asymptotic ratio. A bound for the constant is given by (8.84) as $c < 1$. But, since $\alpha(\alpha+1)$ is a scalar for the weighting function, this can be sharpened to $c \leqslant 1 - \frac{1}{\alpha(\alpha+1)}$. This gives the bound in table 8.16.

Case	Bound	
$\beta = \alpha$	$FFD \leqslant \quad 1 - \frac{1}{\alpha} \quad + \quad CSP_R$	lemma 8.3 (page 81)
$\beta = \alpha + 1$	$FFD \leqslant \quad 1 - \frac{1}{\alpha(\alpha+1)} \quad + \left(\frac{\alpha+3}{\alpha+2} - \frac{2}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R$	§8.10.4.1 (page 137)
$\beta = \alpha + 2$	$FFD \leqslant 1 + \frac{\alpha^2-2}{\alpha(\alpha+1)(\alpha+2)} + \left(\frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R$	(8.86) (page 138)
$\beta = \alpha + 3$	$FFD < \quad 1 + \frac{4}{\alpha+3} \quad + \left(\frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R$	§8.10.4.3 (page 139)
$\beta \geqslant \alpha + 4$	$FFD < \quad \frac{\alpha+2}{\alpha+3} \quad + \quad \frac{\alpha+4}{\alpha+3} \quad Mat$	(8.14) (page 87)

Table 8.16. FFD-bounds for lists with largest item $\varphi \leqslant 1/\alpha$, $\alpha \geqslant 3$ and critical item $x \in \langle\frac{1}{\beta+1}, \frac{1}{\beta}]$

8.10.4.2 $\beta = \alpha + 2$ [13]

This gives a configuration, which is essentially the same as diagram 8.22 (page 108); it consists of four recurrent blocks and three transition bins. As in section 8.8.7, we need to condition on the existence of a bin with largest item $y \in \langle\frac{1-x}{\alpha+1}, \frac{1}{\alpha+1}]$. This gives the following bound.

$$FFD \leqslant 1 + \frac{\alpha^2-2}{\alpha(\alpha+1)(\alpha+2)} + \left(\frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R \tag{8.86}$$

There are two cases to investigate.[14]

8.10.4.2a There is a bin with largest item $y \in \langle\frac{1-x}{\alpha+1}, \frac{1}{\alpha+1}]$ For this case we use the following strengthening of the generic weighting function.

$$W(s) = \begin{cases} \frac{1}{\alpha}, & s \in \langle\frac{1-x}{\alpha}, \frac{1}{\alpha}] \\ \frac{\alpha+3}{(\alpha+1)(\alpha+2)}, & s \in \langle\frac{1-y}{\alpha}, \frac{1-x}{\alpha}] \\ \frac{1}{\alpha+1}, & s \in [y, \frac{1-y}{\alpha}] \\ \frac{1}{\alpha+2}, & s \in [x, y\rangle \end{cases} \tag{8.87}$$

By assumption 8.16 there are no bins with largest item in $\langle\frac{1}{\alpha+1}, \frac{1-y}{\alpha}]$ and with this it is easy to verify that all recurrent bins have a bin-weight of 1. The maximum pattern-weight under this weighting function is determined as $\frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$ (see pages 237–239 for details).

To determine a bound for the constant, first note that there are at most three transition-bins. When an item y is placed in its bin, all previous bins must be filled to a level strictly larger than $1 - y$. This implies that all these bins contain at least α items with size at least y. The second

[13] Using lemma B23 and bound (8.84) one can derive the bound $FFD \leqslant \frac{\alpha+4}{\alpha+2} + \left(\frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R$. This is asymptotically tight when α is odd, but not when α is even.

[14] We note that for $\alpha = 2$ the bounds derived for these cases are the same as those in §8.8.7; viz. (8.44) and (8.46).

and third transition bin contain at least $\alpha+1$ items. This gives the following lower bounds for the weight of the transition bins, and a bound for the constant.

$$\left.\begin{array}{l} W_1 \geqslant \frac{1}{\alpha}+\frac{\alpha-1}{\alpha+1}=1-\frac{\alpha-1}{\alpha(\alpha+1)} \\ W_2 \geqslant \frac{\alpha+3}{(\alpha+1)(\alpha+2)}+\frac{\alpha-1}{\alpha+1}+\frac{1}{\alpha+2}=1 \\ W_3 \geqslant \frac{1}{\alpha+1}+\frac{\alpha}{\alpha+2}=1-\frac{\alpha}{(\alpha+1)(\alpha+2)} \end{array}\right\} \Rightarrow c \leqslant \frac{\alpha-1}{\alpha(\alpha+1)}+0+\frac{\alpha}{(\alpha+1)(\alpha+2)}+1-\frac{1}{\alpha+2}=1+\frac{\alpha^2-2}{\alpha(\alpha+1)(\alpha+2)}$$

Combining the above gives the bound $FFD \leqslant 1+\frac{\alpha^2-2}{\alpha(\alpha+1)(\alpha+2)}+\left(\frac{\alpha+3}{\alpha+2}-\frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}\right) CSP_R$.

8.10.4.2b There is no bin with largest item in $\left\langle\frac{1-x}{\alpha+1}, \frac{1}{\alpha+1}\right]$ For this case we use the following variant of the generic weighting function.

$$W(s)=\begin{cases} \frac{1}{\alpha}, & s \in \left\langle\frac{1-x}{\alpha}, \frac{1}{\alpha}\right] \\ \frac{\alpha+1}{\alpha(\alpha+2)}, & s \in \left\langle\frac{1}{\alpha+1}, \frac{1-x}{\alpha}\right] \\ \frac{1}{\alpha+2}, & s \in \left[x, \frac{1}{\alpha+1}\right] \end{cases} \tag{8.88}$$

The maximum pattern-weight is determined as $1+\frac{\alpha-1}{\alpha(\alpha+2)}$ (see page 239 for details). To determine a bound for the constant, note that there are at most two transition-bins. The first one contains at least α items and the second at least $\alpha+1$. This gives the following bounds for their bin-weight.

$$\left.\begin{array}{l} W_1 \geqslant \frac{1}{\alpha}+\frac{\alpha-1}{\alpha+2}=1-\frac{2\alpha-2}{\alpha(\alpha+2)} \\ W_2 \geqslant \frac{\alpha+1}{\alpha(\alpha+2)}+\frac{\alpha}{\alpha+2}=1-\frac{\alpha-1}{\alpha(\alpha+2)} \end{array}\right\} \Rightarrow c \leqslant \frac{2\alpha-2}{\alpha(\alpha+2)}+\frac{\alpha-1}{\alpha(\alpha+2)}+1-\frac{1}{\alpha+2}=1+\frac{2\alpha-3}{\alpha(\alpha+2)}$$

However, we can sharpen this bound by a more careful analysis. If there is only one transition-bin we have $\min\{W_1, W_2\}=1-\frac{2\alpha-2}{\alpha(\alpha+2)}$ as a lower bound for its weight. This case gives $c \leqslant 1+\frac{\alpha-2}{\alpha(\alpha+2)}$. If there are two transition-bins then there must be at least α items in $\left\langle\frac{1}{\alpha+1}, \frac{1}{\alpha}\right]$ in the first one; when the first item in the second T-bin is placed all previous bins contain only items $> \frac{1}{\alpha+1}$ and must be filled to a level $1-\frac{1}{\alpha}$, this is only possible when they contain at least α items. This gives a tighter lower bound for the weight of the first transition-bin as $W_1 \geqslant \frac{1}{\alpha}+(\alpha-1)\times\frac{\alpha+1}{\alpha(\alpha+2)}=1-\frac{\alpha-1}{\alpha(\alpha+2)}$. Combining this with the bound for W_2, gives the same bound for c; viz. $c \leqslant 1+\frac{\alpha-2}{\alpha(\alpha+2)}$.
This gives the bound $FFD \leqslant 1+\frac{\alpha-2}{\alpha(\alpha+2)}+\left(\frac{\alpha+3}{\alpha+2}-\frac{1}{\alpha(\alpha+2)}\right) CSP_R$, which implies bound (8.86).

8.10.4.3 $\beta=\alpha+3$

Lemma B23 gives an upper bound of $\frac{\alpha+3}{\alpha+2}-\frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$ for the asymptotic ratio. A bound for the constant is given by (8.84) as $c<\frac{\alpha+7}{\alpha+3}$. This gives the bound in table 8.16.

Part III

Applications

Chapter 9

Applications

In this chapter we will show the application of results, derived in the main part of the thesis.

In section 9.1 we will give some bounds for the duality gap of the cutting stock problem, based upon worst-case bounds for bin-packing heuristics. In section 9.2 we will illustrate how one can derive tighter worst-case bin-packing bounds for specific instances. In section 9.3 we will show that a reduction algorithm used in the analysis of FFD is a useful addition to the solution process of the cutting stock problem. In section 9.4 an illustration is given of how one can apply the recurrency concept to a two-dimensional packing problem.

For notational brevity we assume that, for a given list, $\varphi \in \left\langle \frac{1}{\alpha+1}, \frac{1}{\alpha} \right]$ is the largest item in the list, m is the number of different item-types and $\gamma(\mathcal{L})$ is the duality gap for this list.

9.1 Duality gap

Preliminaries Suppose that we have a heuristic H with bound $\forall \mathcal{L}\ H(\mathcal{L}) < c + r\,CSP_R(\mathcal{L})$. This allows us to derive bounds for the duality gap of the associated cutting stock problem. In the following we may assume wlog that we are dealing with a residual CSP. A direct bound for the gap is given by using the heuristic to pack the [residual] list as; $\gamma < c + (r-1)CSP_R(\mathcal{L})$. Another bound is given by rounding up the residual CSP-solution as; $\gamma \leqslant m - CSP_R$. Combining this gives the following bound.

$$\gamma < \frac{c}{r} + \frac{r-1}{r}m \tag{9.1}$$

In particular, for a heuristic with $c + r = 2$ (see also lemma 5.13), this gives $\gamma < 1 + \frac{r-1}{r}(m-2)$.

Heuristic	$R_H^\infty(\varphi)$	$R_H^\infty(\alpha^{-1})$
NF	$1+\frac{\varphi}{1-\varphi}$	$1+\frac{1}{\alpha-1}$
FF	$1+\frac{1}{\alpha}$	$1+\frac{1}{\alpha}$
NFD	$1+\frac{1}{\alpha+1}+\frac{1}{(\alpha+1)(\alpha+2)}+\cdots$	$1+\frac{1}{\alpha+1}+O(\alpha^{-2})$
FFD	$1+\frac{1}{\alpha+2}-\frac{\{1,2\}}{\alpha(\alpha+1)(\alpha+2)}$	$1+\frac{1}{\alpha+2}+O(\alpha^{-3})$

Table 9.1. Asymptotic ratios for different heuristics for lists without 1-items, and largest item $\varphi \in \langle\frac{1}{\alpha+1},\frac{1}{\alpha}]$. Similar tables can be found in the literature.[12, 14]

From a theoretical point of view, the worst-case bounds for the heuristics investigated (NF, FF, NFD and FFD) do not give any significant improvement over the results in chapter 3; viz. $\gamma < 1+\frac{m-2}{\alpha+1}$ (lemma 3.8). This originates from the fact that all these heuristics have [parametric] asymptotic worst-case ratios of the order $1+\alpha^{-1}+o(\alpha^{-1})$ (see table 9.1), which, combined with (9.1) gives a bound for γ of the order $1+O(m/\alpha)$.

However, the bound for the duality gap in lemma 3.8 does not tell us how to construct a heuristic solution that achieves this bound. Surprisingly enough this can be done using the first-fit heuristic. Consider the following hybrid heuristic: solve the LP-relaxation of the cutting stock problem (by means of the Simplex Method) and determine two feasible solutions as follows.

1. Round up the LP-solution.
2. Round down the LP-solution, determine the residual CSP, and pack this using the first-fit heuristic.

Now take as the solution value the minimum of the solution values of 1 and 2. We can use lemma 6.8 and bound (9.1) to show that this gives a heuristic for which $\gamma_H = H - CSP_R < 1+\frac{m-2}{\alpha+1}$, where φ is the largest item in the residual CSP and $\alpha = \lfloor\varphi^{-1}\rfloor$.

The bound in lemma 3.8 leads to the bound $\gamma < m/2$ for $m \geqslant 2$, which is an improvement on corollary 3.6. This can be further improved by using some of the results for the FFD-heuristic as by the following lemma.

9.1 **Lemma** $\gamma(\mathcal{L}) < 1+\frac{2}{11}(OPT-1)$

Proof. *Follows from (8.1) which implies* $OPT < 1+\frac{11}{9}CSP_R$, *and thus* $CSP_R > \frac{9}{11}(OPT-1)$. *Substitution in* $\gamma \equiv OPT - CSP_R$ *now proves the lemma.* □

Since we can assume that the duality gap is maximised for a residual CSP, and the residual has a row dimension not exceeding that of the original CSP we get the following corollaries.

9.2 Corollary $\gamma(\mathcal{L}) < 1 + \frac{2}{11}(m-1)$

9.3 Corollary *Any instance of CSP with at most six different item sizes has the NRU-property.*

We note that there is a limit to how far one can improve bounds of the form $\gamma < 1 + r(m-1)$, due to the instances with a duality gap larger than 1 (table 3.1).

The following corollaries follow directly from lemma 6.8. They are useful since they are in terms of *Mat*, a quantity which is easily calculated.

9.4 Corollary *If* $\varphi \in \left\langle \frac{1}{\alpha+1}, \frac{1}{\alpha} \right]$ *then* $\gamma(\mathcal{L}) < 1 + \frac{Mat-1}{\alpha}$

9.5 Corollary *Any list* $\mathcal{L} \subset \left\langle 0, \frac{1}{\alpha} \right]$ *with* $Mat(\mathcal{L}) \leqslant \alpha + 1$ *has the NRU-property.*

9.2 List-specific bounds

Suppose that we know what item types we are dealing with, but not their frequency. In the setting of the examples in the introduction of chapter 1: customers only order from a prespecified set of bar lengths or the finishing times are for a few specific treatments only.

To illustrate how one can derive tighter bounds in those circumstances, we will consider the following example, which is taken from Pierce [54, p. 115].

$$\text{sizes } \mathbf{d}^{\mathsf{T}} = [64, 60, 48, 45, 33, 32, 16] \text{ on bins of size } L = 215.$$

For any list $\mathcal{L}$ with items solely from this item set we can derive the following.

9.2.1 Duality gap

To apply corollary 3.14 we first calculate $\alpha_i = \lfloor L/d_i \rfloor$ to give $\alpha^{\mathsf{T}} = [3, 3, 4, 4, 6, 6, 13]$. This directly shows that any list $\mathcal{L}$ has the NRU-property. This can be tightened to $\gamma(\mathcal{L}) < 1 + \frac{4 \times 32}{215} \approx 1.5953$ by using (3.7).

9.2.2 First-fit bound

Solving the ratio-program (5.24) for the first-fit heuristic gives the following optimal dual variables $\mathbf{u}^{\mathsf{T}} = \frac{1}{624}[234, 221, 182, 156, 104, 104, 48]$, a value of $\rho^* = \frac{5}{4}$, and thus by (6.10) the following bound.

$$FF < 1 + \tfrac{5}{4} CSP_R \text{ and thus } FF \leqslant \tfrac{3}{4} + \tfrac{5}{4} OPT$$

That this bound is asymptotically tight follows from the instance in diagram 9.1.

9.2.3 Next-fit-decreasing bound

Solving the ratio-program (5.24) for the next-fit decreasing heuristic gives $u_i^* = 1/\alpha_i$, $1 \leqslant i \leqslant 7$, $\mathbf{a}^* = 2\mathbf{e}_2 + 2\mathbf{e}_4$ and a value of $\rho^* = \frac{7}{6}$. Combining this with lemma 7.7 gives the following bounds.

$$NFD < 1\tfrac{5}{12} + \tfrac{7}{6} CSP_R \text{ and thus } NFD \leqslant \tfrac{4}{3} + \tfrac{7}{6} OPT$$

That this bound is asymptotically tight follows from the instance in diagram 9.2. Note that FFD produces the same packing for this worst-case instance.

List $\mathcal{L} = \{ \underbrace{45, \ldots, 45}_{4k}, \underbrace{60, 60, 48}_{4k} \}$ on bins of size 215.

Diagram 9.1. Worst-case instance of first-fit on Pierce-7.

List $\mathcal{L} = \{ \underbrace{60, \ldots, 60}_{12k}, \underbrace{45, \ldots, 45}_{12k} \}$ on bins of size 215.

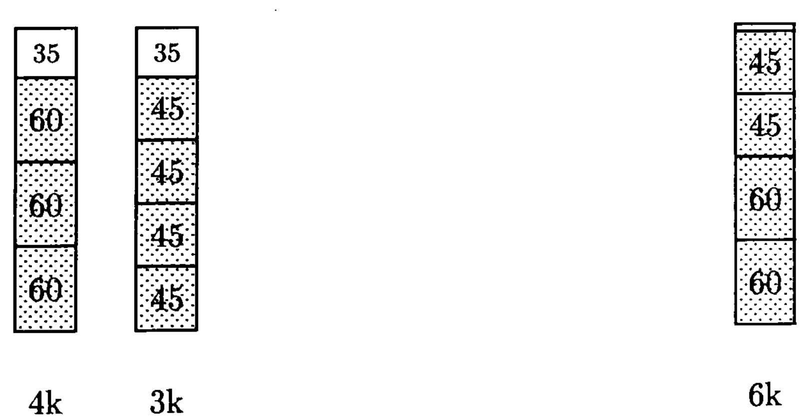


Diagram 9.2. Worst-case instance of next-fit decreasing on Pierce-7.

9.3 Preprocessing by FFD

As shown in section 8.2, the FFD-heuristic can be used to reduce the problem size of a CSP. Moreover, as will be demonstrated, it is actually capable of solving some problems in the literature. This FFD-reduction is particularly useful for problems with relatively large item-sizes. Potential application areas can be found in the paper industry[2,54] and in the metal industry.[59,65]

To test the effect of the FFD-preprocessing algorithm, we have used instances as published in Pierce[54] and Staedtler.[65] In table 9.2 we have taken one instance (Pierce-20) and shown in detail the workings of the reduction algorithm. It solves the instance optimally in 21 steps and produces an optimal solution with value $CSP_l = 1527$ using 21 different patterns. For the instances in table 9.3 we have listed the problem dimension m, the dimension of the reduced problem m_{red}, and the number of bins by which the problem is reduced. Note that $m_{\text{red}} = 0$ means that the reduction procedure completely solves the instance.

Although this reduction procedure may not produce a reduction for all (or even most) practical problems, it is a useful addition to the process of solving a CSP for the following reason. It can reduce the dimension of the problem, that is the number of different item types, and thus reduces the size of the (basis-)matrix that is used in the ensuing simplex-method. The potential computational gain derived from this reduction procedure will certainly outweigh the computational effort needed to implement it.

i	d_i	f_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	96	150	1																				
2	91	20		1																			
3	90	30			1																		
4	89	170				1	1	1															
5	84	80							1														
6	80	420								1	1	1	1	1									
7	75	50													2								
8	73	210														2							
9	72	40															2						
10	69	30								1													
11	67	100									1												
12	66	100							1			1											
13	65	110											1										
14	63	270												1				2					
15	61	30				1																	
16	60	70			1		1																
17	59	310		1				1											2				
18	55	280																		2			
19	50	480	1																		3		
20	49	320																				3	2
pattern			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
wastage			4	0	0	0	1	2	0	1	3	4	5	7	0	4	6	24	32	40	0	3	52
usage			150	20	30	30	40	100	80	30	100	20	110	160	25	105	20	55	95	140	110	106	1

use lemma 8.5 (patterns 1–12) — use lemma 8.6 (patterns 13–18) — use lemma 8.3 (patterns 19–21)

Table 9.2. FFD-preprocessing on Pierce-20: $L = 150$, $CSP_R = 1526\frac{2}{3}$ and $CSP_I = 1527$
The table is a detailed breakdown of the reductions by the algorithm shown in table 8.4 (p. 84).

Problem-ID	m	m_{red}	*FFD*
Pierce-20	20	0	1527
Pierce-30	30	0	2845
Pierce-30RQ	30	0	2575
Pierce-40	40	13	1922
Pierce-75	75	0	3818

Examples taken from Pierce[54]

Problem-ID	m	m_{red}	*FFD*
113816610	6	0	356
130813610	6	5	156
132068610	6	0	356
717839610	8	7	164
717846610	16	10	300

Examples taken from Städtler[65]

Table 9.3. FFD-reduction on problems in literature.

List $\mathcal{L} = \{\underbrace{3,\ldots,3}_{100k}, \underbrace{2,\ldots,2}_{300k}, \underbrace{1,\ldots,1}_{400k}\}$ on a bin of size 5.

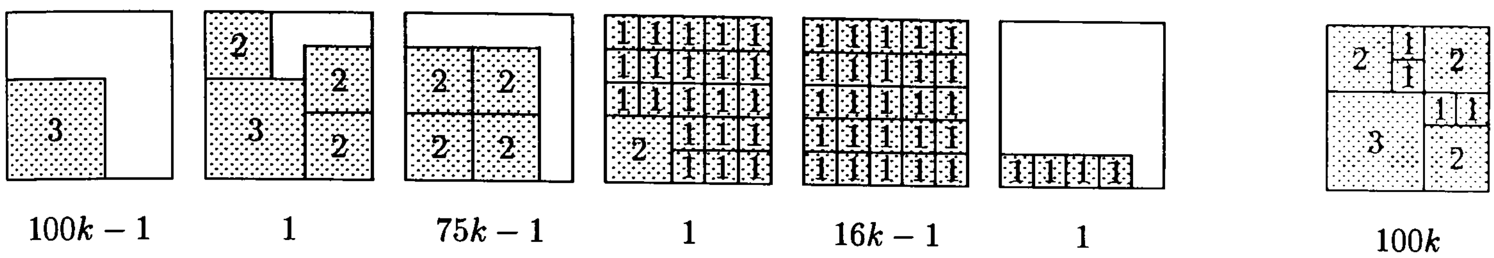


Diagram 9.3. Example of BLNFD square-packing.

9.4 Recurrency for higher dimensional packing problems

To show how one could use the concept of recurrency to derive bounds for higher dimensional packing problems we will analyse a next-fit algorithm to pack squares.

9.4.1 BLNFD-heuristic for square packing

The *Bottom-Left Next-Fit Decreasing (BLNFD)* heuristic takes a list of squares, sorts it into non-increasing order of item size and places each item, in succession, in the current bin, with its sides aligned to the sides of the bin, at the lowest, left-most position. When an item cannot be placed the current bin is closed and a new bin is opened in which this item is placed. As an illustration we give an example in diagram 9.3

Define an i-square to be a square with side $s \in \langle \frac{1}{i+1}, \frac{1}{i}]$. A recurrent bin is a bin which contains i^2 i-squares. This gives $W(s) = 1/i^2$ as a recurrent weighting function, for s an i-square. If the largest square is an α-square, then the ratio problem $\mathcal{R}(\alpha)$ is given by the following program, where $\mathcal{A}$ represents the set of all feasible packings and a_i represents the number of i-squares in a feasible packing. The relaxation $\overline{\mathcal{R}}(\alpha)$ can be obtained by replacing the constraint by one based upon an area argument.

$$\mathcal{R}(\alpha) = \boxed{\begin{array}{ll} \text{Max} & \displaystyle\sum_{i=\alpha}^{\infty} \frac{a_i}{i^2} \\ \text{st} & \mathbf{a} \in \mathcal{A} \end{array}} \leqslant \overline{\mathcal{R}}(\alpha) = \boxed{\begin{array}{ll} \text{Max} & \displaystyle\sum_{i=\alpha}^{\infty} \frac{a_i}{i^2} \\ \text{st} & \displaystyle\sum_{i=\alpha}^{\infty} \frac{a_i}{(i+1)^2} < 1 \\ & a_i \in \mathbb{N} \end{array}} \qquad (9.2)$$

Solving $\mathcal{R}(\alpha)$ is considerably more difficult than the one-dimensional version (see section B.1). This is because one has to take geometrical considerations into account. It is not even known, except for very small values of i, how many i-squares one can pack in a unit-square.[29,34]

Isothetic packing

If only isothetic packings are allowed (i.e. sides of the squares must be parallel to the side of the bin) then there can be at most α^2 α-squares in an [optimal] packing. Adding the constraint $a_\alpha \leqslant \alpha^2$ to the relaxation $\overline{\mathcal{R}}(\alpha)$ in (9.2) gives a bound as follows

$$\overline{\mathcal{R}}(\alpha) < \tfrac{\alpha^2}{\alpha^2} + \left(\tfrac{\alpha+2}{\alpha+1}\right)^2 \left[1 - \tfrac{\alpha^2}{(\alpha+1)^2}\right] = 1 + \tfrac{(2\alpha+1)(\alpha+2)^2}{(\alpha+1)^4} = 1 + \tfrac{2}{\alpha+1} + \tfrac{3}{(\alpha+1)^2} - \tfrac{1}{(\alpha+1)^4}, \qquad (9.3)$$

with values of $2\frac{11}{16}$, $1\frac{80}{81}$, $1\frac{175}{256}$ and $1\frac{324}{625}$ for $\alpha = 1, 2, 3$ and 4 respectively.[1]

Instances

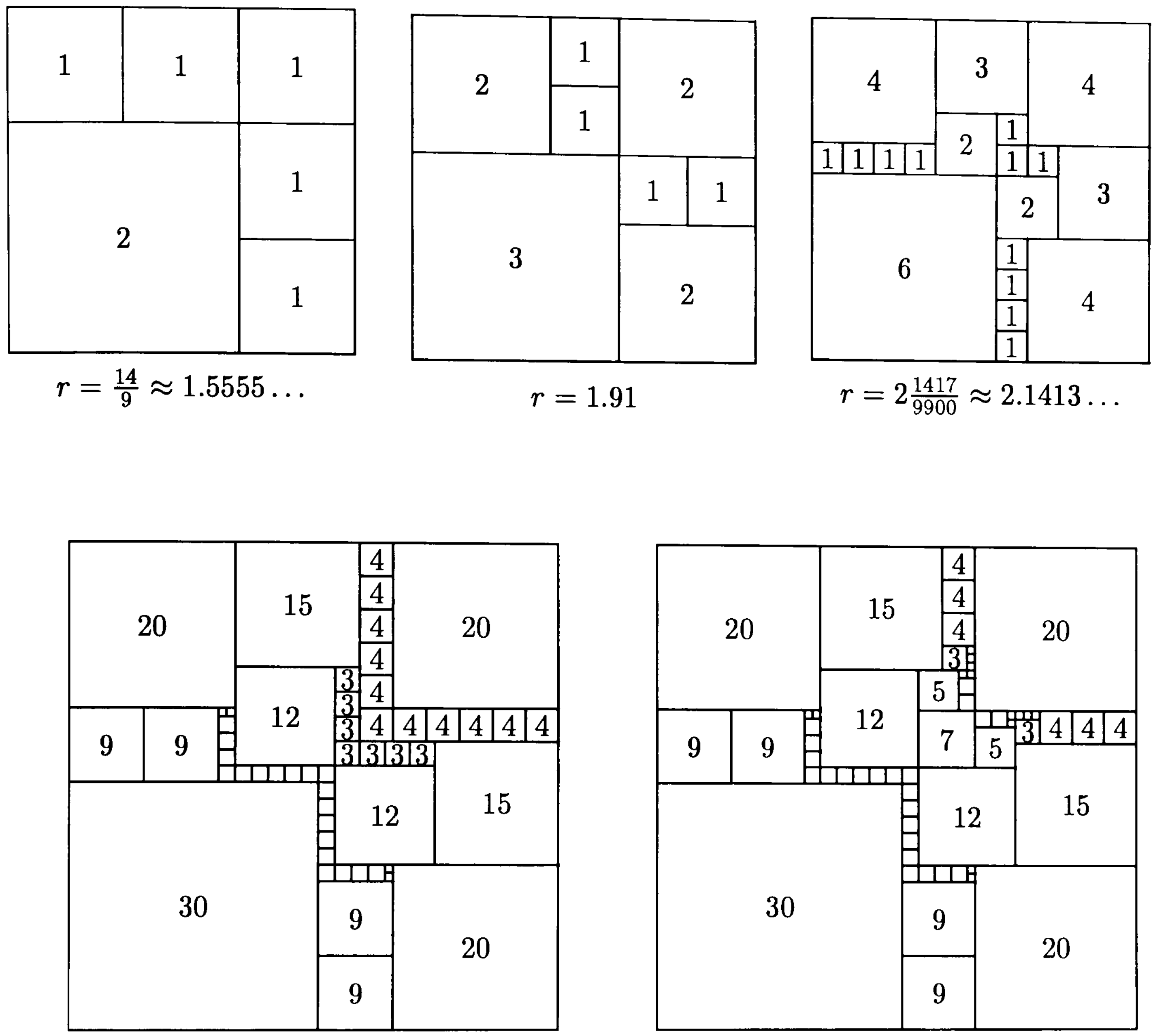
On the next page some examples are given for the BLNFD algorithm on squares. The number of bins used follows from the generalisation of corollary E8, as stated in the note on page 246. The weight of bin j; $\sum a_i/i^2$, with a_i the number of i-squares it contains, is the recurrent ratio that is achieved by the instance listed below.

[1]The value for $\alpha = 1$, viz. $\frac{43}{16}$ coincides with the value given by Coppersmith and Raghavan[16] for their on-line packing algorithm.

We note that for the more general problem of isothetic packing of rectangles by BLNFD one can derive bounds for the recurrent ratio in a similar fashion. If all the rectangles have sides $\leqslant 1/\alpha$ then the following bound holds.

$$\overline{\mathcal{R}}(\alpha) < \tfrac{\alpha^2}{\alpha^2} + \tfrac{(\alpha+1)(\alpha+2)}{\alpha(\alpha+1)} \left[1 - \tfrac{\alpha^2}{(\alpha+1)^2}\right] = 1 + \tfrac{2}{\alpha} + \tfrac{1}{(\alpha+1)^2},$$

with values of $3\frac{1}{4}$, $2\frac{1}{9}$, $1\frac{35}{48}$ and $1\frac{54}{100}$ for $\alpha = 1, 2, 3$ and 4 respectively. Again, the value for $\alpha = 1$, viz. 3.25 coincides with the value given by Coppersmith and Raghavan.



$r = \frac{14}{9} \approx 1.5555\ldots$ $\quad r = 1.91$ $\quad r = 2\frac{1417}{9900} \approx 2.1413\ldots$

$r \approx 2.307587611\ldots$ $\quad r \approx 2.308009897\ldots$

Diagram 9.4. Worst-case examples for the BLNFD-heuristic.

Example 1 $BLNFD = 14k$ and $OPT = 9k$

$$\text{List } \mathcal{L} = \{\ \underbrace{2,\ldots,2}_{9k},\ \underbrace{1,\ldots,1}_{45k}\ \} \text{ on bins of size 3.}$$

Example 2 $BLNFD = 191k$ and $OPT = 100k$

$$\text{List } \mathcal{L} = \{\ \underbrace{3,\ldots,3}_{100k},\ \underbrace{2,\ldots,2}_{300k},\ \underbrace{1,\ldots,1}_{400k}\ \} \text{ on bins of size 5.}$$

Example 3 $BLNFD = 21217k$ and $OPT = 9900k$

$$\text{List } \mathcal{L} = \{\ \underbrace{6,\ldots,6}_{9900k},\ \underbrace{4,\ldots,4}_{29700k},\ \underbrace{3,\ldots,3}_{19800k},\ \underbrace{2,\ldots,2}_{19800k},\ \underbrace{1,\ldots,1}_{108900k}\ \} \text{ on bins of size 11.}$$

9.5 Miscellaneous

In this section we have collected some known results, for which the results derived in this thesis allow an easier derivation.

9.5.1 Divisible CSP

A list is called *weakly divisible* if the item sizes satisfy $d_m|\cdots|d_1$. If in addition, the bin size is a multiple of the largest item size then the list is called *strongly divisible*. Coffman et al.[13] prove that FF is optimal for a strongly divisible list. This can easily be proven using recurrency and bound (6.10) as follows. The strong divisibility means that $\sum x_i > 1 - x_m$ and implies $\sum x_i \geqslant 1$. So that we can take $W(x) = x$ as a recurrent weighting function. Bound (6.10) gives $FF < 1+\sum x_i$ and since $OPT \geqslant \sum x_i$, it follows that $FF = OPT$.

9.5.2 Absolute worst-case bounds

Simchi-Levi[64] gives a result[2] for the absolute worst-case ratio of FFD; $\forall \mathcal{L}\ \frac{FFD(\mathcal{L})}{OPT(\mathcal{L})} \leqslant 1.5$. Although the proof is not difficult, it can be proven in a more simple manner using some of the results derived in chapter 8. Let x be the size of the first item in the last bin.

- If $x > 1/3$ then corollary 8.7 shows that $FFD = OPT$.
- If $x \leqslant 1/3$ we can use (8.14) to give $FFD < 2 + \frac{3}{2}(Mat - 1)$ from which $FFD \leqslant \frac{3}{2}OPT$ follows.

This result is tight, see for example [31, p. 128].

[2]He shows that it holds for XFD, where XFD denotes either the Best-Fit Decreasing or the First-Fit Decreasing heuristic. It actually holds for XFD, where X is any conservative bin-packing heuristic.

Chapter 10

Conclusion

10.1 Conclusion

We have investigated the one-dimensional cutting stock problem, the bin-packing problem, and their relationship.

In the first part of the thesis some characterisations of the instances that maximise the duality gap of the one-dimensional cutting stock problem are given. A [parametric] bound for the duality gap is derived which improves upon the known bounds. For the special case that the sizes of all the items are unit fractions we give an explicit bound for the duality gap. We further give an easily verified condition for a cutting stock problem to have the NRU-property.

In the second part of the thesis we have shown that worst-case bounds for bin-packing problems can be and are better expressed in terms of the LP-relaxation of the associated cutting stock problem. A new formulation of the worst-case ratio for bin-packing heuristics is introduced. This, recurrent ratio, leads to an LP-formulation from which [an upper bound for] the ratio can be determined. The concept of recurrency is put to use in the analysis of the first-fit, next-fit decreasing and first-fit decreasing bin-packing heuristics. For each of these we derive slightly tighter worst-case bounds. For FF we derive tight parametric bounds for the case when there are no 1-items present. For NFD we tighten the known bounds, give a more elegant derivation of the worst-case ratio and construct new worst-case instances. For FFD we tighten the known bounds and show that its worst-case ratio is closely connected to a subset-sum problem on unit fractions.

In the third and final part we put the results of the bounds for bin-packing heuristics to use in the derivation of bounds for the duality gap of the cutting stock problem. We further show how one can derive tight problem-specific worst-case bounds for the first-fit and next-fit decreasing heuristic. Also an illustration of the application of the recurrency concept for the packing of squares by a next-fit algorithm is given.

There are two main conclusions that can be drawn. The first one is that it is unlikely that the question, what the duality gap of the cutting stock problem is, can be settled using the known bin-packing heuristics. As noted in section 9.1, all these heuristics lead to bounds which are functions of the largest item-size, and on the basis of these one cannot even decide whether or not the duality gap is finite. The second conclusion is that the concept of recurrency and the recurrent ratio is well worth a further investigation. Not only does it simplify the analysis leading to a worst-case ratio, but it is of particular use in deriving worst-case bounds for specific instances.

With regard to worst-case performance bounds for bin-packing heuristics there are some recommendations that can be made. These should be expressed in terms of CSP_R (or Mat) instead of OPT. This is even more true for two- and higher dimensional packing heuristics, since these have a larger duality gap. For on-line bin-packing heuristics, as lemma 5.14 indicates, one should [in the first instance] be looking for bounds in terms of Mat.

10.2 Directions for future research

In this section we have listed some topics for future research. These are grouped in line with the chapters in the main body of the text.

10.2.1 The One-dimensional Cutting Stock Problem

(1) Determine an explicit solution for the case $m = 2$.

Although we know that these instances can be solved optimally by linear programming, since they have the RU-property, it would be nice to have an explicit solution in terms of $L, d_{1,2}$ and $f_{1,2}$. This might possibly be used in a constructive heuristic.

(2) Can an optimal solution to CSP_I always be expressed in terms of LP-optimal patterns?

Let $\mathbf{u}^*$ be the optimal dual-multipliers to the linear-programming relaxation CSP_R. Is it true that there is always at least one optimal solution to CSP_I, which consists solely of patterns $\mathbf{a}_j$, such that $\forall j \ \langle \mathbf{a}_j, \mathbf{u}^* \rangle = 1$ holds?

(3) Does lemma 2.7 hold for CSP_I (with the possible exception of one bin)?

(4) For a given instance of CSP let ρ^* be the maximum value of $\sum a_i/\alpha_i$ for any feasible pattern. Is it true that in the optimal solution to CSP_R and CSP_I there is always a pattern active which achieves this value ρ^*? This would give rise to an interesting heuristic for both CSP_R and CSP_I.

(5) Is the least scalar for $\mathbf{u}^*$ always less than or equal to the stock length L?

10.2.2 The Duality gap

It is highly frustrating that even for apparently simple cases we do not know what the maximum duality-gap for CSP is. For instance,

(6) What is the maximum gap for the harmonic CSP?

(7) What is the maximum gap for $OPT = 3$, or even for $m = 3$?

(8) What is the maximum gap for lists $\mathcal{L} \subset \langle \frac{1}{\alpha+2}, \frac{1}{\alpha}]$. In particular,

(a) Is there a heuristic that is optimal for this case?
Recall that for lists $\mathcal{L} \subset \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$ there is a heuristic that is optimal, viz. FFD. Note that FFD is not optimal for all lists on $\langle \frac{1}{\alpha+2}, \frac{1}{\alpha}]$, see the instance in diagram 8.10 (p. 96).

(b) Do these lists have the RU-property?

(c) What is the largest gap when $\alpha = 2$.

The known instances with a duality gap greater than one all pack into a configuration with a singleton bin. That these instances do not have the RU-property relies on the fact that the LP-solution uses only extremal patterns, whereas an optimal solution may require a (wasteful) non-extremal pattern. For the generic CSP (in which trim loss has no value) we may replace the constraint $\mathbf{a} \in \mathcal{A}$ by $\mathbf{a} \in \mathcal{E}$ in formulation (2.1) of CSP_I. This eliminates the singleton bin and gives rise to the following question.

(9) What is the maximum duality-gap if we restrict the patterns in CSP_I to extremal patterns only?

10.2.3 The Harmonic Cutting Stock Problem

(10) A class of harmonic CSPs, worth investigating is when all sizes are of the form p_i^{-k}, where p_i is the i^{th} prime number. For instance take a number m, with $p_i^m \leqslant n$ and construct a list of $(p_i - 1)$ items of size $p_i^{-1}, \ldots, p_i^{-m}$. This list contains no CTDs and has

$$CSP_R(n) = \sum_{i=1}^{\pi(n)} \sum_k \tfrac{p_i - 1}{p_i^k} = \sum_{i=1}^{\pi(n)} \left(1 - p_i^{-m_i}\right) \leqslant (1 - n^{-1})\pi(n)$$

where $m_i = \lfloor \ln n / \ln p_i \rfloor$

(11) An interesting variation on the harmonic heuristic (where one rounds up the sizes of the items to the nearest unit-fraction), is suggested by the results in chapter 4.

1) Round up all item sizes; $x \to 1/\lfloor 1/x \rfloor$
2) Reduce the problem, if there are i such that $f_i \geqslant \text{sp}(i)$, by aggregating $\text{sp}(i)$ i-items into one $(i/\text{sp}(i))$-item.
3) Pack the resulting list by FFD.

This heuristic outperforms FFD on lists of unit fractions (see for instance diagram E.1). What is the least value of c for this heuristic, such that $H \leqslant c + CSP_R$ holds for harmonic lists? This value is at least $10 - (S(23) - 2/10) \approx 1.238$ (see table 4.1, p. 30).

10.2.4 Bin-packing Heuristics

(12) Given the [known] results for the standard bin-packing heuristics (table 9.1), would it be true that all heuristics of complexity $O(n)$ have worst-case performance ratio of order $1 + O(\alpha^{-1})$ for lists on $\langle 0, \alpha^{-1}]$.

(13) What is a good (low complexity) heuristic that has worst-case performance $1 + O(\alpha^{-2})$ for lists on $\langle 0, \alpha^{-1}]$. Such heuristics would be extremely useful in limiting the candidates for, if not solving the maximum duality-gap problem of CSP.

(14) What are necessary and sufficient conditions on a bin-packing heuristic for the definitions of the recurrent- and asymptotic worst-case ratio to coincide?

10.2.5 The First-Fit Heuristic

(15) What is the worst-case ratio in terms of the scalar of a list?

Lemma 5.15 suggests that the asymptotic ratio for FF should be expressed in terms of the scalar for $\mathcal{L}$, in order to derive a worst-case bound which is achievable.

10.2.6 The Next-Fit Decreasing Heuristic

(16) Extend the parametrisation of the NFD-bounds to the case where one also has a lower bound for the smallest item-size [as well as an upper bound for the largest].

This will give a definite value for the ratio, rather than an infinite series.

(17) Sharpen the bound in lemma 7.7 to derive tight parametric bounds for $|NFD(\mathcal{L}) - \sum_{i=1}^{m} f_i/\alpha_i|$.

10.2.7 The First-Fit Decreasing Heuristic

The most obvious questions are, do the following two bounds hold?

(18) $FFD \leqslant \frac{22}{27} + \frac{11}{9} CSP_R$, for $FFD \geqslant 2$

(19) $FFD \leqslant \frac{2}{3} + \frac{11}{9} OPT$

The analysis in chapter 8 and the failure to find any counter-examples leads to the following conjecture. We note that, under this conjecture, for $\varphi = \frac{1}{\alpha}$, the instances in tables B.6 and B.7 show that this [the implied] bound is the best possible. We note further that to prove or disprove this conjecture the bin-configuration in diagram 8.44, needs further investigation.

(20) $FFD(\mathcal{L}) < 2 + R^{\infty}_{FFD}(\varphi)(OPT(\mathcal{L}) - 1)$, for all lists $\mathcal{L} \subset \langle 0, \varphi]$ **(Conjecture)**

There are some other worst-case bounds, which are worth investigating.

(21) It should be possible to prove the bound $FFD < 2 + \frac{\alpha+3}{\alpha+2}(CSP_R - 1)$ for lists $\mathcal{L} \subset \langle 0, 1/\alpha]$. This would allow a slight sharpening of the bound for the duality gap, to $\gamma < 1 + \frac{m-2}{\alpha+3}$.

The answer to the following questions would possibly allow the use of less sophisticated weighting functions and simplify the analysis, necessary to prove the worst-case bounds.

(22) Are there other (simple) methods to rule out certain configurations for FFD? In particular $(FFD, OPT) = (13, 10), (24, 19)$ and $(18, 14)$ (see tables 8.11 and 8.14).

(23) It seems that a better way of tackling the problem of deriving worst-case bounds for FFD is to condition on the existence and the size of the recurrent items.

Note that, for $x \in \langle \frac{1}{5}, \frac{1}{4}]$ bins with largest item in $\langle \frac{1-x}{3}, \frac{1}{3}]$ are essential to achieve the $\frac{11}{9}$-ratio. Similar, for $x \in \langle \frac{2}{11}, \frac{1}{5}]$ bins with largest item in $\langle \frac{1-x}{3}, \frac{1}{3}]$, and bins with largest item in $\langle \frac{1-x}{4}, \frac{1}{4}]$ are essential to achieve the $\frac{11}{9}$-ratio.

For $x \in \langle \frac{1}{5}, \frac{1}{4}]$ we have the recurrent items z, v, y and x (see diagram 8.17). The conditions that are necessary to achieve a ratio of $\frac{11}{9}$ are $z+y+x \leqslant 1$ and $2v+2x \leqslant 1$. For $x \in \langle \frac{2}{11}, \frac{1}{5}]$ we have the recurrent items z, v, u and y (see diagram 8.44). The conditions that are necessary to achieve a ratio of $\frac{11}{9}$ are $z+u+y \leqslant 1$ and $2u+2y \leqslant 1$.

(24) An interesting observation is that the $\frac{11}{9}$-configuration for $x \in \langle \frac{2}{11}, \frac{1}{5}]$ is basically the same as the one for $x \in \langle \frac{1}{5}, \frac{1}{4}]$ with an item in $\langle \frac{2}{11}, \frac{1}{5}]$ added. This leads one to think that it should be possible to derive a bound for the case $x \in \langle \frac{2}{11}, \frac{1}{5}]$ based upon the results for $x \in \langle \frac{1}{5}, \frac{1}{4}]$

(25) For $x \geqslant \frac{2}{11}$ we might be able to use the fact that $3x \geqslant 2 \times \frac{1-x}{3}$ and the dominance that follows from it to eliminate the case that there are 1-items $\geqslant 3x$.

(26) For lists with no 1-items, strengthen the bounds (8.80) and (8.81)

This can be achieved by choosing a stronger weighting function and subsequently using the set-packing approach. A possible candidate for the weighting function is obtained by replacing $W(s) = \frac{\beta-1}{\beta}\frac{1}{i}$ in (8.82) by $W(s) = \frac{\beta+1}{\beta}\frac{1}{i+1}$. A necessary condition for this to be a stronger weighting function is $\frac{\beta+1}{\beta}\frac{1}{i+1} \geqslant \frac{\beta-1}{\beta}\frac{1}{i}$ that is $\forall i\ 2i \geqslant \beta - 1$, which is true when $\beta \leqslant 2\alpha + 1$ holds. Note that this is exactly the condition for the generic weighting function to be recurrent!!

List $\mathcal{L} = \{\ 3 \times 3,\ \underbrace{2 \times 2, \ldots, 2 \times 2}_{6}\ \}$ on a square of size 6×6

Diagram 10.1. Instance with two item types and a gap larger than 1

10.2.8 Two-dimensional problems

(27) How many i-squares can one pack in a unit-square? (see §9.4.1)

The only known value is for $i = 1$; one can pack at most one 1-square in a unit-square.

(28) What is the recurrent ratio for isothethic square-packing by a NFD-heuristic?

In particular, is $\mathcal{R}(\alpha)$ as defined in (9.2) greedy solvable, as in the one-dimensional case?

(29) Is there an equivalent of (3.1) for the two-dimensional cutting stock problem?

In section 3.3 we have seen that any instance of the one-dimensional cutting stock problem with at most two different item-types has the RU-property. This does not hold in general for the two-dimensional problem, as is illustrated by the instance in diagram 10.1. The proof of corollary 3.7 relies on the elementary relationship (3.1), which does not necessarily hold for higher dimensional cutting problems.

10.2.9 Complexity issues

(30) Complexity of harmonic CSP.

In particular instances with $Mat < 2$.

(31) Is PARTITION restricted to unit fractions NP-complete?

If so, then CSP is NP-complete, even when we know the optimal solution to the LP-relaxation.

Part IV

Appendices

Appendix A

Integrality assumption for CSP

A1 Theorem ***(Integrality assumption for CSP)*** *For any instance of the cutting stock problem, defined by CSP$(m, L, \mathbf{d}, \mathbf{f})$ we may assume that L and $\mathbf{d}$ are integral.*

Proof. *Given L and $\mathbf{d}$, define ε^* by the following subset-sum problem;*

$$L + \varepsilon^* \quad = \quad \text{Min} \quad \left\{ \sum a_i d_i \mid \sum a_i d_i > L, \; a_i \in \mathbb{N} \right\}.$$

Further define

$$\begin{aligned} \alpha &= \max_i \lfloor L/d_i \rfloor. \\ L' &= L + \tfrac{\alpha}{\alpha+1}\varepsilon \text{ , by choosing } \varepsilon \in \left\langle 0, \varepsilon^* \right] \text{ such that } L' \text{ is rational.} \\ d_i' &= d_i + \tfrac{1}{\alpha+1}\varepsilon_i \text{ , by choosing } \varepsilon_i \in \left[0, \varepsilon\right] \text{ such that } d_i' \text{ is rational.} \end{aligned}$$

Finally define $\mathcal{A} = \left\{\mathbf{a} \in \mathbb{N}^m \mid \sum a_i d_i \leqslant L\right\}$ and $\mathcal{A}' = \left\{\mathbf{a} \in \mathbb{N}^m \mid \sum a_i d_i' \leqslant L'\right\}$. We can now prove that $\mathcal{A} = \mathcal{A}'$ as follows.

(1) $\mathbf{a} \in \mathcal{A} \Rightarrow \mathbf{a} \in \mathcal{A}'$

$\sum a_i d_i' = \sum a_i d_i + \sum \frac{\varepsilon_i}{\alpha+1} a_i \leqslant L + \frac{\alpha}{\alpha+1}\varepsilon \leqslant L'$, *since* $\varepsilon_i \leqslant \varepsilon$ *and* $\sum a_i \leqslant \alpha$.

(2) $\mathbf{a} \in \mathcal{A}' \Rightarrow \mathbf{a} \in \mathcal{A}$

$\sum a_i d_i = \sum a_i d_i' - \sum \frac{\varepsilon_i}{\alpha+1} a_i \leqslant L + \frac{\alpha}{\alpha+1}\varepsilon < L + \varepsilon \Rightarrow \sum a_i d_i \leqslant L$, *since there is no pattern with a length in the interval* $\left\langle L, L+\varepsilon\right\rangle$.

Since $\mathcal{A} = \mathcal{A}'$ one can use the rationals $\mathbf{d}'$ and L' to define the CSP. We can now scale the generating equation $\sum a_i d_i' \leqslant L'$ to integers. □

The equivalence of CSP and BPP implies the following corollary.

A2 Corollary ***(Integrality assumption for BPP)*** *For any instance of the bin-packing problem, defined by a particular list $\mathcal{L}$, we may assume that the sizes of the items to be packed are rationals.*

Appendix B

Integer Programming Models

All the programs studied are variations of the (unbounded) *knapsack problem.*

$$\mathcal{K}P \equiv \max\left\{\sum p_i\delta_i \mid \sum w_i\delta_i \leqslant 1,\ \delta_i = 0,1\right\}$$

A useful heuristic in this context is the *greedy* heuristic. Suppose that we have items with sizes c.q. weights w_i and profits p_i. The *relative profit* or *bang-for-buck* value is defined as $\rho(i) = p_i/w_i$. The greedy heuristic works as follows. (A more detailed description can be found in the literature on knapsack problems[49, 67]).

1. sort the items such that $\rho(1) \geqslant \rho(2) \geqslant \cdots \geqslant \rho(n)$ and set $W = 0$.
2. For $i := 1$ to n do

   If $W + w_i \leqslant 1$ Then $\delta_i := 1$, $W := W + w_i$ Else $\delta_i := 0$.

Obviously, the greedy heuristic provides a feasible solution (and thus a lower bound) to the knapsack problem. But it also provides an upper bound. If j is the index of the first item that cannot be placed, $P = \sum_{i=1}^{j-1} p_i$ and $W = \sum_{i=1}^{j-1} w_i$ then $\mathcal{K}P \leqslant P + \rho(j) \times (1 - W)$. The upper bounds based upon the greedy heuristic and the relative profit are used to rule out certain solutions as not optimal. A program is said to be *greedy solvable* if a greedy heuristic returns an optimal solution.

In section B.1 a knapsack-type program is defined which is closely related to the asymptotic ratio of the next-fit decreasing algorithm. The solution of this gives rise to a sequence of numbers, the properties of which are studied in section B.2. On the basis of these results we define a weighting function for NFD. Its corresponding ratio problem is studied in section B.3.

In section B.4 we introduce a program which is closely linked to the asymptotic ratio of the first-fit decreasing algorithm. On the basis of its solution we derive instances which provide lower bounds for the FFD-ratio. Finally, in section B.7 we investigate how to find sizes that generate a knapsack polytope. This is used in chapter 8 to find a set of sizes that will pack into a certain bin-configuration.

B.1 NFD ratio-problem

For $\alpha \in \mathbb{N}^+$ define the following integer programs.

$$\mathcal{R}(\alpha) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i+1} < 1 \\ & a_i \in \mathbb{N} \end{array}} \quad \text{and} \quad \widetilde{\mathcal{R}}(\alpha) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\infty} \frac{a_i}{i+1} < \frac{1}{\alpha} \\ & a_i \in \mathbb{N} \end{array}} \tag{B1}$$

Note that in both programs the items are arranged in decreasing 'bang-for-buck' value;

$$\rho(i) = \frac{\text{profit}(i)}{\text{weight}(i)} = \frac{i+1}{i} = 1 + \frac{1}{i} \Rightarrow \rho(1) > \rho(2) > \cdots \tag{B2}$$

B1 Lemma *$\widetilde{\mathcal{R}}(\alpha)$ is greedy solvable.*

Proof. *By showing that the highest relative-profit item, a_α is active.*

i) $\widetilde{\mathcal{R}}(\alpha) \geqslant \frac{1}{\alpha} + \frac{1}{\alpha(\alpha+1)} = \frac{1}{\alpha} \times \frac{\alpha+2}{\alpha+1}$, *since a feasible solution is given by* $\frac{1}{\alpha+1} + \frac{1}{\alpha(\alpha+1)+1} < \frac{1}{\alpha}$.

ii) $a^*_\alpha = 1$, *since* $a^*_\alpha = 0$ *implies* $\widetilde{\mathcal{R}}(\alpha) < \max_{i>\alpha} \rho(i) \times \frac{1}{\alpha} = \frac{\alpha+2}{\alpha+1} \times \frac{1}{\alpha}$, *which contradicts i).*

The fact that $a^*_\alpha = 1$ *implies* $a^*_{\alpha+1}, \ldots, a^*_{\alpha(\alpha+1)} = 0$, *since* $\frac{1}{\alpha} - \frac{1}{\alpha+1} = \frac{1}{\alpha(\alpha+1)}$. *From this it follows that* $\widetilde{\mathcal{R}}(\alpha) = \frac{1}{\alpha} + \widetilde{\mathcal{R}}(\alpha(\alpha+1))$ *and the lemma thus follows by induction.* □

B2 Lemma *$\mathcal{R}(\alpha)$ is greedy solvable.*

Proof. *By expressing $\mathcal{R}$ in terms of $\widetilde{\mathcal{R}}$.*

i) $\mathcal{R}(\alpha) \geqslant 1 + \frac{1}{\alpha+1} + \frac{1}{(\alpha+1)(\alpha+2)}$, *since a feasible solution is given by* $\frac{\alpha}{\alpha+1} + \frac{1}{\alpha+2} + \frac{1}{(\alpha+1)(\alpha+2)+1} < 1$.

ii) $a^*_\alpha = \alpha$, *since* $a^*_\alpha \leqslant \alpha - 1$ *implies* $\mathcal{R}(\alpha) < \frac{\alpha-1}{\alpha} + \max_{i>\alpha} \rho(i) \times (1 - \frac{\alpha-1}{\alpha+1}) = \frac{\alpha-1}{\alpha} + \frac{\alpha+2}{\alpha+1} \times \frac{2}{\alpha+1} = 1 + \frac{1}{\alpha+1} + \frac{1}{(\alpha+1)(\alpha+2)} - \frac{2}{\alpha(\alpha+1)^2(\alpha+2)}$, *which contradicts i).*

Since $a^*_\alpha = \alpha$ *we have* $\mathcal{R}(\alpha) = 1 + \widetilde{\mathcal{R}}(\alpha+1)$ *and the lemma follows directly from lemma B1* □

The greedy solvability allows us to express the value of the programs (and with it the asymptotic ratios of NFD) as recurrence relationships or alternatively, as an infinite series, as shown by the following corollaries.

B3 Corollary $\mathcal{R}(\alpha) = 1 + \widetilde{\mathcal{R}}(\alpha+1)$

B4 Corollary $\widetilde{\mathcal{R}}(\alpha) = \frac{1}{\alpha} + \widetilde{\mathcal{R}}(\alpha(\alpha+1))$

B5 Corollary $\mathcal{R}(\alpha) = 1 + \frac{1}{\alpha+1} + \frac{1}{(\alpha+1)(\alpha+2)} + \frac{1}{(\alpha+1)(\alpha+2)[(\alpha+1)(\alpha+2)+1]} + \cdots$

When truncating the series for $\mathcal{R}$ or $\widetilde{\mathcal{R}}$, we need a bound for the remaining terms. A lower bound is provided by a feasible solution, as in the first clause of the lemmas. An upper bound is provided by $\rho(\alpha) \times 1$ and $\rho(\alpha) \times \frac{1}{\alpha}$, respectively. This gives the following corollaries.

$\alpha\backslash n$	0	1	2	3	4	5
1	1	2	6	42	1806	3263442
2	2	3	12	156	24492	599882556
3	3	4	20	420	176820	31265489220
4	4	5	30	930	865830	749662454730
5	5	6	42	1806	3263442	10650056950806
6	6	7	56	3192	10192056	103878015699192
7	7	8	72	5256	27630792	763460694178056
8	8	9	90	8190	67084290	4500302031888390
9	9	10	110	12210	149096310	22229709804712410
10	10	11	132	17556	308230692	95006159799029556

Table B.1. Sequence $\{b_n\}$ for different values of α

B6 Corollary $\frac{\alpha+2}{\alpha+1} < \mathcal{R}(\alpha) < \frac{\alpha+1}{\alpha}$

B7 Corollary $\frac{1}{\alpha} + \frac{1}{\alpha(\alpha+1)} < \widetilde{\mathcal{R}}(\alpha) < \frac{1}{\alpha} + \frac{1}{\alpha^2}$

We will finish this section with some observations.

• The 'finite' version of $\mathcal{R}$ is not necessarily greedy solvable. This is illustrated by the following example; $\max\left\{\sum_{i=1}^{4} \frac{a_i}{i} \mid \sum_{i=1}^{4} \frac{a_i}{i+1} < 1 \text{ and } a_i \in \mathbb{N}\right\}$. The greedy solution is $\mathbf{a}^\mathsf{T} = [1,1,0,0]$ with value 3/2, whereas the optimal solution is $\mathbf{a}^\mathsf{T} = [1,0,1,1]$ with value 19/12.

• Relationship to FF-ratio. Without affecting the solution space of $\mathcal{R}(1)$ we can add the implicit constraint $a_1 \leqslant 1$. If we now relax the strict inequality in $\mathcal{R}(1)$ to a 'less or equal', the resulting program is easily solved. It has a solution value of $1.7 = 1 + \frac{1}{2} + \frac{1}{5}$ for the items $\frac{1}{2}, \frac{1}{3}, \frac{1}{6}$, and is the asymptotic ratio for first-fit. For $\alpha \geqslant 2$ we can relax the '$<$' in $\mathcal{R}(\alpha)$ to '$\leqslant$' and the resulting program has $\frac{\alpha+1}{\alpha}$ as a solution value, which is the parametric asymptotic ratio for first-fit, for lists where all items have a size $\leqslant \frac{1}{\alpha}$.

B.2 Doubly exponential sequence $\{b_n\}$

For a fixed, positive integer α the sequence $\{b_n\}$ of integers is defined as follows.

$$b_0 = \alpha\ ,\quad b_1 = \alpha + 1 \quad \text{and} \quad b_{n+1} = b_n(b_n + 1) \tag{B3}$$

This sequence is one of several studied by Aho and Sloane.[1] An equivalent definition, by Golomb[35] is

$$b_0 = \alpha\ ,\quad b_{n+1} = \prod_{i=0}^{n} (b_i + 1) \tag{B4}$$

The sequence is readily generated; see table B.1. From this table and property B8 (easily proved by induction), we see that it very quickly yields extremely large numbers. Properties B9–B12 are used in chapter 7 to construct worst-case instances for the NFD-heuristic.

B8 Property $b_n \geqslant (b_1)^{2^{n-1}} \geqslant 2^{2^{n-1}}$, *for* $n \in \mathbb{N}^+$

B9 Property ***(successive divisibility)*** $b_1 \mid b_2 \mid \cdots \mid b_n$

B10 Property ***(minimal scalar)*** b_N *is the minimal scalar for* $\{\frac{1}{b_1}, \ldots, \frac{1}{b_{N-1}}\}$ *and* $\{\frac{1}{b_1+1}, \ldots, \frac{1}{b_{N-1}-1}\}$

B11 Property $\forall 1 \leqslant n \leqslant N \ \sum_{i=n}^{N-1} \frac{1}{b_i+1} = \frac{1}{b_n} - \frac{1}{b_N}$

Proof. *This is a direct consequence of the way the sequence* $\{b_n\}$ *is derived.*

$$\sum_{i=n}^{N-1} \frac{1}{b_i+1} = \sum_{i=n}^{N-1} \left(\frac{1}{b_i} - \frac{1}{b_i(b_i+1)}\right) = \sum_{i=n}^{N-1} \frac{1}{b_i} - \sum_{i=n}^{N-1} \frac{1}{b_{i+1}} = \frac{1}{b_n} - \frac{1}{b_N}$$

For $N = n$ *the lemma is obvious and for* $N \geqslant n+1$ *the lemma follows from the above.* □

This lemma allows the following property as a corollary.

B12 Property $\forall N \in \mathbb{N}^+ \ \frac{\alpha}{\alpha+1} + \sum_{n=1}^{N-1} \frac{1}{b_n+1} = 1 - \frac{1}{b_N}$

For $\alpha = 1$, this property simplifies to $\sum_{n=0}^{N-1} \frac{1}{b_n+1} = 1 - \frac{1}{b_N}$. Golomb[35] mentions the conjecture that the closest approximation to 1 from below, which is a sum of N *unit fractions* is given, for every value of N, by $\frac{1}{2} + \frac{1}{3} + \frac{1}{7} + \frac{1}{43} + \frac{1}{1807} + \cdots + \frac{1}{b_{N-1}+1}$; viz. the sequence $\{b_n + 1\}$ for $\alpha = 1$.

Relationship to the NFD-ratio problem

We can now express the solution value of program $\mathcal{R}(\alpha)$ in terms of the elements of the sequence $\{b_n\}$ (the corresponding values of the active variables are; $a_\alpha = \alpha$ and $a_{b_1} = a_{b_2} = \cdots = a_{b_n} = \cdots = 1$).

$$\mathcal{R}(\alpha) = 1 + \sum_{n=1}^{\infty} \frac{1}{b_n} \tag{B5}$$

In order to bound the error when we truncate this sum, we rewrite it as

$$\mathcal{R}(\alpha) = 1 + \sum_{n=1}^{N} \frac{1}{b_n} + \widetilde{\mathcal{R}}(b_{N+1}) \tag{B6}$$

Now use lemma B13 and property B8 to derive an upper bound for the 'error'-term $\widetilde{\mathcal{R}}(b_{N+1})$

$$\widetilde{\mathcal{R}}(b_{N+1}) < b_N^{-2} \leqslant (\alpha+1)^{-2^N} \leqslant 2^{-2^N} \tag{B7}$$

This shows that the sum $\sum \frac{1}{b_n}$ converges very rapidly; the number of significant bits roughly doubles with every term. The speed of convergence can be verified in table B.2.

In the derivation of [a bound for] the constant in the worst-case bound for NFD we will need the following lemmas.

$\alpha\backslash N$	0	1	2	3	4	5
1	1	1.5	1.666...	1.690476190	1.691029900	1.691030207
2	1	1.333...	1.416666...	1.423076923	1.423117753	1.423117754
3	1	1.25	1.3	1.302380952	1.302386608	1.302386608
4	1	1.2	1.233333333	1.234408602	1.234409757	1.234409757
5	1	1.1666666...	1.190476190	1.191029900	1.191030207	1.191030207
6	1	1.142857143	1.160714286	1.161027569	1.161027667	1.161027667
7	1	1.125	1.138888888	1.139079148	1.139079184	1.139079184
8	1	1.111111	1.122222222	1.122344322	1.122344337	1.122344337
9	1	1.1	1.109090909	1.109172809	1.109172816	1.109172816
10	1	1.090909091	1.098484848	1.098541809	1.098541812	1.098541812

Table B.2. Convergence of $\mathcal{R}_N(\alpha) = 1 + \sum_{n=1}^{N} \frac{1}{b_n}$ for different values of α

B13 Lemma $\forall 1 \leqslant n \leqslant N \quad \sum_{i=n+1}^{N} \frac{1}{b_i} < \sum_{i=n+1}^{\infty} \frac{1}{b_i} = \tilde{\mathcal{R}}(b_{n+1}) < \frac{1}{b_{n+1}} + \left(\frac{1}{b_{n+1}}\right)^2 < \frac{1}{b_{n+1}-1} \leqslant \left(\frac{1}{b_n}\right)^2$

Proof. *The first part follows directly from corollary B7. Using the recurrence definition (B3) for* b_{n+1} *and substituting* b *for* b_n *it is easily verified that* $\frac{1}{b(b+1)} + \left(\frac{1}{b(b+1)}\right)^2 < \frac{1}{b(b+1)-1} \leqslant \left(\frac{1}{b}\right)^2$ *holds for* $b \geqslant 1$, *which proves the last part of the lemma.* □

B14 Lemma $\forall 1 \leqslant n \leqslant N \quad \sum_{i=n}^{N-1} \left(\frac{1}{b_i+1}\right)^2 = \left(\frac{1}{b_N}\right)^2 - \left(\frac{1}{b_n}\right)^2 + 2 \sum_{i=n+1}^{N} \frac{1}{b_i} < \left(\frac{1}{b_N}\right)^2 + \left(\frac{1}{b_n}\right)^2$

Proof. *Straightforward algebra.*

$$
\begin{aligned}
\sum_{i=n}^{N-1} \left(\frac{1}{b_i+1}\right)^2 &= \sum_{i=n}^{N-1} \left(\frac{1}{b_i} - \frac{1}{b_i(b_i+1)}\right)^2 = \sum_{i=n}^{N-1} \left(\frac{1}{b_i^2} - \frac{2}{b_i{}^2(b_i+1)} + \frac{1}{b_i^2(b_i+1)^2}\right) \\
&= \sum_{i=n}^{N-1} \frac{1}{b_i^2} - 2 \sum_{i=n}^{N-1} \left(\frac{1-b_i}{b_i{}^2} + \frac{1}{b_i+1}\right) + \sum_{i=n}^{N-1} \frac{1}{b_{i+1}^2} \\
&= \sum_{i=n}^{N-1} \frac{1}{b_i^2} - 2 \sum_{i=n}^{N-1} \frac{1}{b_i^2} + 2 \sum_{i=n}^{N-1} \frac{1}{b_i} - 2 \sum_{i=n}^{N-1} \frac{1}{b_i+1} + \sum_{i=n+1}^{N} \frac{1}{b_i^2} \\
\text{use lemma B11} \quad &= \frac{1}{b_N^2} - \frac{1}{b_n^2} + 2 \sum_{i=n}^{N-1} \frac{1}{b_i} - 2 \left(\frac{1}{b_n} - \frac{1}{b_N}\right) \\
&= \frac{1}{b_N^2} - \frac{1}{b_n^2} + 2 \sum_{i=n+1}^{N} \frac{1}{b_i} \\
\text{use lemma B13} \quad &< \frac{1}{b_N^2} - \frac{1}{b_n^2} + \frac{2}{b_n^2} = \frac{1}{b_N^2} + \frac{1}{b_n^2}.
\end{aligned}
$$

For $N = n$ *the lemma is obvious and for* $N \geqslant n+1$ *the lemma follows from the above.* □

B.3 Finite NFD IP-models

Given a list $\mathcal{L} \subset \langle 0, \frac{1}{\alpha}]$, we can calculate $b_0 = \alpha$ and $b_1, b_2, \ldots$ as defined by (B3) or (B4), and determine the least N such that $\mathcal{L} \subset \langle \frac{1}{b_N}, \frac{1}{\alpha}]$ holds. In the following we will assume $N \geqslant 2$ (otherwise we have the trivial case $\mathcal{L} \subset \langle \frac{1}{\alpha+1}, \frac{1}{\alpha}]$). We define the following weighting function as a strengthening of (7.6);

$$W(x) = \begin{cases} \rho_0 x, & & x \in \langle \frac{1}{\rho_0 \alpha}, \frac{1}{\alpha}] \\ \frac{1}{\alpha}, & & x \in \langle \frac{1}{\alpha+1}, \frac{1}{\rho_0 \alpha}] \\ \rho_n x, & 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{\rho_n b_n}, \frac{1}{b_n}] \\ \frac{1}{b_n}, & 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{b_n+1}, \frac{1}{\rho_n b_n}] \\ \rho_n x, & 1 \leqslant n < N \text{ and } & x \in \langle \frac{1}{b_{n+1}}, \frac{1}{b_n+1}] \end{cases} \tag{B8}$$

where ρ_n are constants satisfying

$$1 + \frac{1}{\alpha+1} \leqslant \rho_0 \leqslant 1 + \frac{1}{2\alpha} + \frac{\alpha+1}{2} \sum_{i=2}^{N} \frac{1}{b_i} \tag{B9}$$

$$1 + \frac{1}{b_n+1} \leqslant \rho_n \leqslant 1 + b_n \sum_{i=n+1}^{N} \frac{1}{b_i}, \quad 1 \leqslant n \leqslant N-1. \tag{B10}$$

Note that in order to ensure the existence of ρ_0 for all values of α, we need the requirement $N \geqslant 2$. Now define the following ratio programs for $1 \leqslant n \leqslant N-1$, and the weighting function W as defined in (B8).

$$\mathcal{P}(0) = \boxed{\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i \leqslant 1 \\ & \frac{1}{b_N} < x_i \leqslant \frac{1}{\alpha} \end{array}} \quad \text{and} \quad \mathcal{P}(n) = \boxed{\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i \leqslant \frac{1}{b_n} \\ & \frac{1}{b_N} < x_i \leqslant \frac{1}{b_n} \end{array}} \tag{B11}$$

Before we tackle these programs we will give some properties of the constants ρ_n. Property B15 follows directly from substitution of the bound, given in lemma B13, in (B9) and (B10). Property B16, by comparing the upper bounds in property B15 with the lower bounds in (B9) and (B10), using the fact that $\frac{1}{b_n+1} \geqslant \frac{1}{b_{n+1}}$ holds for all n. Note that property B15 implies that the intervals $\langle \frac{1}{b_n+1}, \frac{1}{\rho_n}\frac{1}{b_n}]$ are non-vacuous. We will now derive bounds for $\mathcal{P}(n)$ in the following lemmas. These prove that $\mathcal{P}(n) \leqslant \widetilde{\mathcal{R}}(b_n)$, and in particular $\mathcal{P}(0) \leqslant \mathcal{R}(\alpha)$.

B15 Property $\forall 0 \leqslant n \leqslant N-1 \quad \rho_n < 1 + \frac{1}{b_n}$

B16 Property $\rho_0 > \rho_1 > \cdots > \rho_{N-1} > 1$

The weighting function W as defined in (B8) has the following properties. Property B17 is easily verified, using the properties of ρ_n, since we only need to check for the end-points of the intervals. Property B18 makes a similar statement about the relative profit of an item; viz. $W(x)/x$. Property B19 shows that the weighting function is recurrent.

B17 Property $W(x)$ *is non-decreasing*

B18 Property $\rho_n \leqslant W(x)/x < \rho_{n-1}$, *for* $1 \leqslant n \leqslant N-1$ *and* $x \in \langle \frac{1}{b_{n+1}}, \frac{1}{b_n}]$

Proof. *Follows directly from* $W(x)/x < 1 + \frac{1}{b_n} \leqslant 1 + \frac{1}{b_{n-1}+1} \leqslant \rho_{n-1}$ □

B19 Property $W(x) \geqslant 1/\lfloor 1/x \rfloor$

Proof. *The property is obvious for all* n *and* $x \in \langle \frac{1}{b_n+1}, \frac{1}{b_n}]$. *Now suppose that* $x \in \langle \frac{1}{i+1}, \frac{1}{i}] \subset \langle \frac{1}{b_{n+1}}, \frac{1}{b_n+1}]$. *Then,* $\lfloor 1/x \rfloor \times W(x) = i\rho_n x > \frac{i}{i+1}\rho_n \geqslant \frac{i}{i+1}\frac{b_n+2}{b_n+1} \geqslant 1$ *and proves the lemma.* □

B20 Lemma $\forall 1 \leqslant n \leqslant N-1 \quad \sum_{i=n}^{N-1} \frac{1}{b_i} \leqslant \mathcal{P}(n) \leqslant \sum_{i=n}^{N} \frac{1}{b_i}$

Proof. *To prove the lower bound we construct a list consisting of one item each of size* $x_i = \frac{b_N}{b_N - b_n} \times \frac{1}{b_i+1}$, *for* $n \leqslant i \leqslant N-1$. *This list represents a feasible solution since* $\sum_{i=n}^{N-1} x_i = \frac{b_N}{b_N-b_n} \sum_{i=n}^{N-1} \frac{1}{b_i+1} = \frac{b_N}{b_N-b_n}(\frac{1}{b_n} - \frac{1}{b_N}) = \frac{1}{b_n}$, *by lemma B11. It is easily verified that* $\frac{1}{b_N} < x_{N-1} < \cdots < x_n \leqslant \frac{1}{b_n}$ *holds for all items, and thus* $W(x_i) \geqslant 1/\lfloor 1/x_i \rfloor \geqslant \frac{1}{b_i}$ *by property B19. This proves the lower bound.*

To prove the upper bound we perform a case analysis on the number of items that are active in the interval $\langle \frac{1}{b_n+1}, \frac{1}{\rho_n}\frac{1}{b_n}]$. *Since* $\sum x_i \leqslant \frac{1}{b_n}$ *there can be at most one such item. If there is none then* $\mathcal{P}(n) = \sum_i W(x_i) \leqslant \sum \max_i(\frac{W(x_i)}{x_i})x_i = \rho_n \sum x_i \leqslant \rho_n \frac{1}{b_n} \leqslant \sum_{i=n}^{N} \frac{1}{b_i}$ *by property B18.*

Now assume that there is one such item active, say of size x_1. *If* $n = N-1$ *then* $\mathcal{P}(n) = \frac{1}{b_n}$. *Therefore assume* $n \leqslant N-2$. *A bound for* $\mathcal{P}(n)$ *is then given by;*

$$\begin{aligned}\mathcal{P}(n) &= W(x_1) + \sum_{i\neq 1} W(x_i) \leqslant \frac{1}{b_n} + \sum_{i \neq 1} \max_i \left(\frac{W(x_i)}{x_i}\right) x_i \leqslant \frac{1}{b_n} + \rho_{n+1} \sum_{i \neq 1} x_i \leqslant \frac{1}{b_n} + \rho_{n+1}\left(\frac{1}{b_n} - x_1\right) \\ &< \frac{1}{b_n} + \rho_{n+1}\left(\frac{1}{b_n} - \frac{1}{b_n+1}\right) = \frac{1}{b_n} + \rho_{n+1}\frac{1}{b_{n+1}} \leqslant \frac{1}{b_n} + \sum_{i=n+1}^{N} \frac{1}{b_i} = \sum_{i=n}^{N} \frac{1}{b_i}\end{aligned}$$

This proves the upper bound and thus completes the proof of the lemma. □

B21 Lemma $1 + \sum_{i=1}^{N-1} \frac{1}{b_i} \leqslant \mathcal{P}(0) \leqslant 1 + \sum_{i=1}^{N} \frac{1}{b_i}$

Proof. *To prove the lower bound we construct a list consisting of* α *items of size* $x_0 = \frac{b_N}{b_N-1} \times \frac{1}{\alpha+1}$, *and one item each of size* $x_i = \frac{b_N}{b_N-1} \times \frac{1}{b_i+1}$, *for* $1 \leqslant i \leqslant N-1$. *This list represents a feasible solution since* $\sum_{i=0}^{N-1} x_i = \frac{b_N}{b_N-1} \times (\frac{\alpha}{\alpha+1} + \sum_{i=1}^{N-1} \frac{1}{b_i+1}) = \frac{b_N}{b_N-1}(1 - \frac{1}{b_N}) = 1$, *by property B12, and* $\frac{1}{b_N} < x_{N-1} < \cdots < x_0 \leqslant \frac{1}{\alpha}$, *so that both constraints in* $\mathcal{P}(0)$ *are satisfied. It is easily verified that* $\frac{1}{b_i+1} < x_i \leqslant \frac{1}{b_i}$ *holds for all items, and thus* $W(x_i) \geqslant 1/\lfloor 1/x_i \rfloor \geqslant \frac{1}{b_i}$, *using property B19. This proves the lower bound.*

To prove the upper bound we perform a case analysis on the number of items a^*, *that are active in the interval* $\langle \frac{1}{\alpha+1}, \frac{1}{\rho_0 \alpha}]$ *in an optimal solution. A bound is given by* $\mathcal{P}(0) \leqslant \frac{a^*}{\alpha} + \rho_0(1 - \frac{a^*}{\alpha+1}) = \rho_0 + (\frac{1}{\alpha} - \frac{\rho_0}{\alpha-1})a^*$. *Note that the multiplicand of* a^* *is strictly positive by property B15. If* $a^* \leqslant \alpha - 1$ *then* $\mathcal{P}(0) \leqslant 1 - \frac{1}{\alpha} + \rho_0 \frac{2}{\alpha-1}$ *holds. A comparison, using (B9), will show that the bound* $\mathcal{P}(0) \leqslant 1 + \sum_{i=1}^{N} \frac{1}{b_i}$ *also holds.*

This leaves the case $a^* = \alpha$. There are no items active with a size in $\langle \frac{1}{\rho_0 \alpha}, \frac{1}{\alpha}]$, *and the residual length is strictly less than* $\frac{1}{\alpha+1}$. *This means that all other active items have a size in* $\langle \frac{1}{b_N}, \frac{1}{b_1}]$, *so that* $\mathcal{P}(0) = 1 + \mathcal{P}(1)$. *The upper bound for this case now follows directly from lemma B20.*

This proves the upper bound and thus completes the proof of the lemma. □

B.4 FFD ratio-problem

For $\alpha, \beta \in \mathbb{N}$ and $\beta \geqslant \alpha \geqslant 2$, define the following programs.[1]

$$\mathcal{D}_\alpha(\beta) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\beta-1} \frac{a_i}{i} + \frac{\beta-1}{\beta} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i} + \frac{a_\beta}{\beta} \\ \text{st} & \sum_{i=\alpha}^{\beta-1} \left(\frac{1-x}{i}\right)^+ a_i + \sum_{i=\alpha}^{\beta-2} \left(\frac{1}{i+1}\right)^+ b_i + x a_\beta \leqslant 1 \\ & x \in \left\langle \frac{1}{\beta+1}, \frac{1}{\beta}\right] \text{ and } a_i, b_i \in \mathbb{N} \end{array}} \tag{B12}$$

If a_β is the only active variable, it is easily verified that the constraint implies $a_\beta \leqslant \beta$ and thus that $\mathcal{D}_\alpha(\beta) = 1$. In particular $\mathcal{D}_\alpha(\alpha) = 1$.

Now assume that $\beta \geqslant \alpha+1$ and that at least one other variable is active. This implies that we can replace the '$\leqslant$'-sign by a '$<$'-sign. Corollary C4 tells us that we can solve the resulting program as the maximum of two related programs: one for each of the extremal values of x. Now scale the constraint by dividing it by $(1-x)$ and then substitute the extremal values $x = \frac{1}{\beta}$ and $x = \frac{1}{\beta+1}$, to give

$$\text{constraint for } x = \tfrac{1}{\beta+1}: \quad \text{st} \quad \sum_{i=\alpha}^{\beta-1} \frac{a_i}{i} + \frac{\beta+1}{\beta} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i+1} + \frac{1}{\beta} a_\beta < \frac{\beta+1}{\beta} \tag{B13}$$

$$\text{constraint for } x = \tfrac{1}{\beta}: \quad \text{st} \quad \sum_{i=\alpha}^{\beta-1} \frac{a_i}{i} + \frac{\beta}{\beta-1} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i+1} + \frac{1}{\beta-1} a_\beta < \frac{\beta}{\beta-1} \tag{B14}$$

Substituting the constraints in (B12) gives the following two programs;

$$\mathcal{R}_\alpha(\beta) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\beta} \frac{a_i}{i} + \frac{\beta-1}{\beta} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\beta} \frac{a_i}{i} + \frac{\beta+1}{\beta} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i+1} < \frac{\beta+1}{\beta} \\ & a_i, b_i \in \mathbb{N} \end{array}}, \quad \tilde{\mathcal{R}}_\alpha(\beta) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\beta-1} \frac{a_i}{i} + \frac{\beta-1}{\beta} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\beta-1} \frac{a_i}{i} + \frac{\beta}{\beta-1} \sum_{i=\alpha}^{\beta-2} \frac{b_i}{i+1} < \frac{\beta}{\beta-1} \\ & a_i, b_i \in \mathbb{N} \end{array}} \tag{B15}$$

Note that in the second program we have set $a_\beta = 0$, since $a_{\beta-1}$ dominates a_β; both have weight $\frac{1}{\beta-1}$, but the former has a larger profit; $\frac{1}{\beta-1} > \frac{1}{\beta}$. Note further that in the first program we can extend the summation from $\sum_{i=\alpha}^{\beta-2}$ to $\sum_{i=\alpha}^{\beta-1}$. This does not alter the value of the program, since $b_{\beta-1}$ is dominated by a_β. Now increase the term $\frac{\beta-1}{\beta}$ in the objective function to $\frac{\beta}{\beta+1}$, which gives the second program for $\beta+1$, and thus shows that $\mathcal{R}_\alpha(\beta) \leqslant \tilde{\mathcal{R}}_\alpha(\beta+1)$.

We can express the solution value of (B12) as $\mathcal{D}_\alpha(\beta) = \max\{\mathcal{R}_\alpha(\beta), \tilde{\mathcal{R}}_\alpha(\beta)\}$. For the range of parameters that we are interested in we can relate this to program (B24) (page 171), which is a *subset-sum problem* on unit fractions.

[1]Note that this is the ratio problem under the generic weighting function for a list with largest item $\varphi \leqslant 1/\alpha$ and critical item $x \in \left\langle \frac{1}{\beta+1}, \frac{1}{\beta}\right]$.

B22 Lemma *For $\alpha = 2$ and $\beta \leqslant 6$ the solution value of (B12) is given by the following table.*

β	2	3	4	5	6
$\mathcal{D}_2(\beta)$	1	$\frac{7}{6}$	$\frac{29}{24}$	$\frac{71}{60}$	$\frac{6}{5}$

Proof. *Solving each case explicitly gives the following tables.*

	$\mathbf{a}^*$	$\mathbf{b}^*$	value
$\mathcal{R}_2(2)$	[2]	–	1
$\mathcal{R}_2(3)$	[1, 2]	–	$\frac{7}{6}$
$\mathcal{R}_2(4)$	[0, 2, 2]	[0]	$\frac{7}{6}$
$\mathcal{R}_2(5)$	[0, 1, 1, 3]	[0, 0]	$\frac{71}{60}$
$\mathcal{R}_2(6)$	[0, 0, 0, 1, 1]	[2, 0, 0]	$\frac{6}{5}$

and

	$\mathbf{a}^*$	$\mathbf{b}^*$	value
$\tilde{\mathcal{R}}_2(2)$	–	–	–
$\tilde{\mathcal{R}}_2(3)$	[1]	–	1
$\tilde{\mathcal{R}}_2(4)$	[1, 1]	[1]	$\frac{29}{24}$
$\tilde{\mathcal{R}}_2(5)$	[0, 2, 1]	[0, 1]	$\frac{71}{60}$
$\tilde{\mathcal{R}}_2(6)$	[0, 1, 1, 1]	[1, 0, 0]	$\frac{6}{5}$

The lemma now follows easily, since $\mathcal{D}_2(\beta) = \max\{\mathcal{R}_2(\beta), \tilde{\mathcal{R}}_2(\beta)\}$. □

B23 Lemma *For $3 \leqslant \alpha \leqslant \beta \leqslant 2\alpha + 1$ the solution value of (B12) is given by*

$$\mathcal{D}_\alpha(\beta) = \begin{cases} 1, & \beta = \alpha \\ \frac{\alpha+3}{\alpha+2} - \frac{2}{\alpha(\alpha+1)(\alpha+2)}, & \beta = \alpha + 1 \\ \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}, & \beta = \alpha + 2 \\ \frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}, & \beta = \alpha + 3 \\ \mathcal{S}_\alpha(\beta - 1), & \beta \geqslant \alpha + 4 \end{cases} \tag{B16}$$

Proof. *The various cases follow from the lemmas B26, B27 and B28.*

- $\mathcal{D}_\alpha(\alpha) = 1$
- $\mathcal{D}_\alpha(\alpha+1) = \max\{\mathcal{R}_\alpha(\alpha+1), \tilde{\mathcal{R}}_\alpha(\alpha+1)\} = \max\{\mathcal{S}_\alpha(\alpha+1), \mathcal{S}_\alpha(\alpha)\} = \mathcal{S}_\alpha(\alpha+1)$
- $\mathcal{D}_\alpha(\alpha+2) = \max\{\mathcal{R}_\alpha(\alpha+2), \tilde{\mathcal{R}}_\alpha(\alpha+2)\} = \max\{\mathcal{S}_\alpha(\alpha+2), \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}\} = \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}$
- $\mathcal{D}_\alpha(\alpha+3) = \max\{\mathcal{R}_\alpha(\alpha+3), \tilde{\mathcal{R}}_\alpha(\alpha+3)\} = \max\{\mathcal{S}_\alpha(\alpha+3), \mathcal{S}_\alpha(\alpha+2)\} = \mathcal{S}_\alpha(\alpha+2)$

For $\alpha + 4 \leqslant \beta \leqslant 2\alpha + 1$:

- $\mathcal{D}_\alpha(\beta) = \max\{\mathcal{R}_\alpha(\beta), \tilde{\mathcal{R}}_\alpha(\beta)\} = \max\{\mathcal{S}_\alpha(\beta), \mathcal{S}_\alpha(\beta-1)\} = \mathcal{S}_\alpha(\beta-1)$

Substituting the values for $\mathcal{S}_\alpha(\beta)$ (table B.4) proves the lemma. □

This allows the following corollaries.

B24 Corollary $\max_{\alpha \leqslant \beta \leqslant 2\alpha+1} \mathcal{D}_\alpha(\beta) = \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}$

B25 Corollary $\max_{\substack{3 \leqslant \alpha \leqslant \beta \leqslant 2\alpha+1 \\ \beta \neq \alpha+2}} \mathcal{D}_\alpha(\beta) = \frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$

We will now derive the necessary lemmas.

B26 Lemma *$\mathcal{R}_\alpha(\beta) = \mathcal{S}_\alpha(\beta)$, for $3 \leqslant \alpha \leqslant \beta \leqslant 2\alpha + 1$*

Proof. *First note that $\mathcal{S}_\alpha(\beta)$ is a lower bound for $\mathcal{R}_\alpha(\beta)$ by choosing $\mathbf{b} = \mathbf{0}$. An upper bound follows from the observation that the relative profits, see (B15), are 1 for the items a_i and $\frac{\beta-1}{\beta+1}\frac{i-1}{i}$ for the items b_i. The latter do not exceed 1 for $\beta \leqslant 2\alpha + 1$. This implies that the bound in the constraint is an upper bound for the value of the objective function, and thus that $\mathcal{R}_\alpha(\beta) < \frac{\beta+1}{\beta}$.*

- *For $\beta \leqslant \alpha + 1$ the lemma is obvious.*
- *For $\beta = \alpha+2$ we note that $\lambda = \alpha(\alpha+1)(\alpha+2)/\gcd\{\alpha, 2\}$ is a scalar for the constraint so that $\mathcal{R}_\alpha(\alpha+2) \leqslant \frac{\alpha+3}{\alpha+2} - 1/\lambda = \mathcal{S}_\alpha(\alpha+2)$, which proves the lemma for $\beta = \alpha + 2$.*

Let $\mathbf{b}^*$ be an optimal solution. We will show that $\mathbf{b}^* = 0$ must hold, from which the lemma follows trivially, since $\mathcal{R}$ and $\mathcal{S}$ then define the same program.

- *For $\beta \geqslant \alpha + 3$ we can derive an upper bound by substituting the constraint in the objective function. The lower bound follows from observation 4 (page 171).*

$$\frac{\beta+1}{\beta} - \frac{2}{(\beta-2)(\beta-1)\beta} \leqslant \mathcal{S}_\alpha(\beta) \leqslant \mathcal{R}_\alpha(\beta) < \frac{\beta+1}{\beta} + \sum_{i=\alpha}^{\beta-2}\left[\frac{\beta-1}{\beta}\frac{1}{i} - \frac{\beta+1}{\beta}\frac{1}{i+1}\right] b_i^* \tag{B17}$$

So that the following must hold.

$$\frac{-2}{(\beta-2)(\beta-1)\beta} < \sum_{i=\alpha}^{\beta-2}\left[\frac{\beta-1}{\beta}\frac{1}{i} - \frac{\beta+1}{\beta}\frac{1}{i+1}\right] b_i^* \Rightarrow \sum_{i=\alpha}^{\beta-2}\left[\frac{2i-\beta+1}{i(i+1)}\right] b_i^* < \frac{2}{(\beta-2)(\beta-1)} \tag{B18}$$

For $2i \geqslant \beta + 1$ we have $\frac{2i-\beta+1}{i(i+1)} \geqslant \frac{2i-\beta+1}{(\beta-2)(\beta-1)} \geqslant \frac{2}{(\beta-2)(\beta-1)}$, which means that $b_i^* = 0$ for $2i \geqslant \beta + 1$. We now only need to consider b_i^* for $2i \leqslant \beta \leqslant 2\alpha + 1$, which are the cases $i = \alpha$ and $\beta = 2\alpha, 2\alpha + 1$.
For $i = \alpha$ and $\beta = 2\alpha + 1$ item b_α has a profit of $\frac{2\alpha}{2\alpha+1}\frac{1}{\alpha} = \frac{2}{2\alpha+1}$ and a weight of $\frac{2\alpha+2}{2\alpha+1}\frac{1}{\alpha+1} = \frac{2}{2\alpha+1}$. This item is dominated by item $a_{(2\alpha+1)}$, so that we may assume that $b_\alpha^* = 0$.
For $i = \alpha$ and $\beta = 2\alpha$ (B18) gives the condition $\frac{b_\alpha^*}{\alpha(\alpha+1)} < \frac{2}{(2\alpha-2)(2\alpha-1)} \Rightarrow b_\alpha^* < \frac{\alpha(\alpha+1)}{(\alpha-1)(2\alpha-1)}$. For $b_\alpha^* \geqslant 1$ to hold we must have $\alpha \leqslant 3$; for all other values of α we have $b_\alpha^* = 0$. This leaves only the case $(\alpha, \beta) = (3, 6)$ to investigate. Solving $\mathcal{R}_3(6)$ explicitly for this case gives $\mathbf{a}^* = [0, 1, 2, 3]$ and $\mathbf{b}^* = \mathbf{0}$.
Ergo for $\beta \geqslant \alpha + 3$ we have $\mathbf{b}^* = \mathbf{0}$, which proves the lemma. □

B27 Lemma *$\widetilde{\mathcal{R}}_\alpha(\beta) = \mathcal{S}_\alpha(\beta - 1)$, for $\alpha \geqslant 3$, $\alpha + 1 \leqslant \beta \leqslant 2\alpha + 1$ and $\beta \neq \alpha + 2$*

Proof. *First note that $\mathcal{S}_\alpha(\beta - 1)$ is a lower bound for $\widetilde{\mathcal{R}}_\alpha(\beta)$ by choosing $\mathbf{b} = \mathbf{0}$. An upper bound follows from the observation that the relative profits, see (B15), are 1 for the items a_i and $\left(\frac{\beta-1}{\beta}\right)^2 \frac{i+1}{i}$ for the items b_i. The latter are strictly less than 1 for $\beta \leqslant 2\alpha + 1$. This implies that the bound in the constraint is an upper bound for the value of the objective function, and thus that $\widetilde{\mathcal{R}}_\alpha(\beta) < \frac{\beta}{\beta-1}$.*

- *For $\beta \leqslant \alpha + 1$ the lemma is obvious.*

Let $\mathbf{b}^*$ be an optimal solution. We will show that $\mathbf{b}^* = 0$ must hold, from which the lemma follows trivially, since $\widetilde{\mathcal{R}}$ and $\mathcal{S}$ then define the same program.

• *For $\beta \geqslant \alpha+3$ we can derive an upper bound by substituting the constraint in the objective function. The lower bound follows from observation 4 (page 171).*

$$\frac{\beta}{\beta-1} - \frac{2}{(\beta-3)(\beta-2)(\beta-1)} \leqslant \mathcal{S}_\alpha(\beta-1) \leqslant \widetilde{\mathcal{R}}_\alpha(\beta) < \frac{\beta}{\beta-1} + \sum_{i=\alpha}^{\beta-2}\left[\frac{\beta-1}{\beta}\frac{1}{i} - \frac{\beta}{\beta-1}\frac{1}{i+1}\right] b_i^* \tag{B19}$$

So that the following must hold.

$$\frac{-2}{(\beta-3)(\beta-2)(\beta-1)} < \sum_{i=\alpha}^{\beta-2}\left[\frac{\beta-1}{\beta}\frac{1}{i} - \frac{\beta}{\beta-1}\frac{1}{i+1}\right] b_i^* \Rightarrow \sum_{i=\alpha}^{\beta-2}\left[\frac{(2\beta-1)i-(\beta-1)^2}{\beta i(i+1)}\right] b_i^* < \frac{2}{(\beta-3)(\beta-2)} \tag{B20}$$

For $i=\beta-2$ we have $\frac{\beta^2-3\beta+1}{\beta(\beta-2)(\beta-1)}b^*_{(\beta-2)} < \frac{2}{(\beta-3)(\beta-2)} \Rightarrow b^*_{(\beta-2)} < \frac{2\beta(\beta-1)}{(\beta-3)(\beta^2-3\beta+1)}$. For $b^*_{(\beta-2)} \geqslant 1$ to hold β must satisfy $\beta(\beta-2)(\beta-6) < 3$ which is the case for $\beta \in \{2,\dots,6\}$. Combining this with $\alpha+3 \leqslant \beta \leqslant 2\alpha+1$ and $\alpha \geqslant 3$ gives the special case $(\alpha,\beta) = (3,6)$. For all the other values of α and β we have $b^*_{(\beta-2)} = 0$. Solving $\widetilde{\mathcal{R}}_3(6)$ explicitly for case $(\alpha,\beta) = (3,6)$ gives $\mathbf{a}^* = [1,1,3]$ and $\mathbf{b}^* = \mathbf{0}$.

For $i \leqslant \beta-3$ and $2i \geqslant \beta+1$ (which implies $\beta \geqslant 7$) we have $\frac{(2\beta-1)i-(\beta-1)^2}{\beta i(i+1)} \geqslant \frac{(2\beta-1)i-(\beta-1)^2}{\beta(\beta-2)(\beta-1)} \geqslant \frac{5\beta-3}{2\beta(\beta-3)(\beta-2)}$. This is strictly larger than $\frac{2}{(\beta-3)(\beta-2)}$ for $\beta \geqslant 3$. So that $b_i^* = 0$ must hold for $i \leqslant \beta-3$ and $2i \geqslant \beta+1$.

For $2i \leqslant \beta \leqslant 2\alpha+1$ we only have the cases $i=\alpha$ and $\beta = 2\alpha, 2\alpha+1$.

For $i=\alpha$ and $\beta = 2\alpha+1$ item b_α has a profit of $\frac{2\alpha}{2\alpha+1}\frac{1}{\alpha} = \frac{2}{2\alpha+1}$ and weight $\frac{2\alpha+1}{2\alpha}\frac{1}{\alpha+1}$, so that this item is dominated by item $a_{(2\alpha+1)}$, and we may assume $b_\alpha^* = 0$.

For $i=\alpha$ and $\beta=2\alpha$ the inequality (B20) gives $\frac{3\alpha-1}{2\alpha^2(\alpha+1)}b_\alpha^* < \frac{1}{(2\alpha-3)(\alpha-1)}$. The only values of α for which $b_\alpha^* \geqslant 1$ can hold are $\alpha = 2,3$. For all other values of α we may assume $b_\alpha^* = 0$. But $\alpha+3 \leqslant \beta \leqslant 2\alpha+1$ rules out $\alpha=2$, so that we are left with the special case $(\alpha,\beta) = (3,6)$ for which $\mathbf{b}^* = \mathbf{0}$ holds.

Ergo for all cases we have $\mathbf{b}^* = \mathbf{0}$, which proves the lemma. □

B28 Lemma $\widetilde{\mathcal{R}}_\alpha(\alpha+2) = \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}$

Proof. *A lower bound follows from $[a_\alpha, a_{(\alpha+1)}, b_\alpha] = [\alpha-1,1,1]$ with value $\frac{\alpha+2}{\alpha+1} - \frac{1}{\alpha(\alpha+2)}$; the upper bound by substituting the constraint in the objective function:*

$$\frac{\alpha+2}{\alpha+1} - \frac{1}{\alpha(\alpha+2)} \leqslant \widetilde{\mathcal{R}}_\alpha(\alpha+2) < \frac{\alpha+2}{\alpha+1} + \left[\frac{\alpha+1}{\alpha+2}\frac{1}{\alpha} - \frac{\alpha+2}{\alpha+1}\frac{1}{\alpha+1}\right] b_\alpha^*. \tag{B21}$$

So that the following must hold, which shows that the only possible values for b_α^* are $\{0,1\}$.

$$\frac{-1}{\alpha(\alpha+2)} < \frac{-(\alpha^2+\alpha-1)}{\alpha(\alpha+1)^2(\alpha+2)}b_\alpha^* \Rightarrow b_\alpha^* < \frac{(\alpha+1)^2}{\alpha^2+\alpha-1} < 2 \tag{B22}$$

If $b_\alpha^* = 0$ then $\widetilde{\mathcal{R}}_\alpha(\alpha+2) = \mathcal{S}_\alpha(\alpha+1) = 1 + \frac{\alpha-1}{\alpha(\alpha+1)} = \frac{\alpha+2}{\alpha+1} - \frac{1}{\alpha(\alpha+1)}$.

If $b_\alpha^* = 1$ then we can tighten the (resulting) constraint to

$$\frac{a_\alpha}{\alpha} + \frac{a_{(\alpha+1)}}{\alpha+1} + \frac{\alpha+2}{\alpha+1}\frac{b_\alpha}{\alpha+1} < \frac{\alpha+2}{\alpha+1} \quad \text{and} \quad b_\alpha = 1 \Rightarrow \frac{a_\alpha}{\alpha} + \frac{a_{(\alpha+1)}}{\alpha+1} \leqslant \frac{\alpha^2+\alpha-1}{\alpha(\alpha+1)} \tag{B23}$$

which gives a value of $\widetilde{\mathcal{R}}_\alpha(\alpha+2) = \frac{\alpha^2+\alpha-1}{\alpha(\alpha+1)} + \frac{\alpha+1}{\alpha+2}\frac{1}{\alpha} = \frac{\alpha+2}{\alpha+1} - \frac{1}{\alpha(\alpha+2)}$.

So, for both possible values of b_α^* we have a value of $\frac{\alpha+2}{\alpha+1} - \frac{1}{\alpha(\alpha+2)} = \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+1)(\alpha+2)}$, which proves the lemma. □

B.5 FFD Subset-Sum Problem

For $\alpha, \beta \in \mathrm{N}$ and $\beta \geqslant \alpha \geqslant 2$, define the following subset-sum problems on unit fractions. Their importance is given by their relationship to the asymptotic ratio of FFD.

$$\mathcal{S}_\alpha(\beta) = \boxed{\begin{array}{ll} \text{Max} & \sum_{i=\alpha}^{\beta} \frac{a_i}{i} \\ \text{st} & \sum_{i=\alpha}^{\beta} \frac{a_i}{i} < \frac{\beta+1}{\beta} \\ & a_i \in \mathrm{N} \end{array}} \qquad \text{(B24)}$$

The solution for a small range of parameters is given in table B.3. We make the following observations with regard to (B24).

1) An upper bound is given by

$$\mathcal{S}_\alpha(\beta) \leqslant 1 + 1/\beta - 1/\lambda_\alpha(\beta), \qquad \text{(B25)}$$

where $\lambda_\alpha(\beta)$ is defined as the least common multiple of $\{\alpha, \alpha+1, \ldots, \beta\}$.[2]

2) For $\beta \leqslant \alpha + 2$ this bound is tight (see table B.4).

3) For $\beta \geqslant \alpha + 3$ a feasible solution to $\mathcal{S}_\alpha(\beta - 1)$ is given by $a_{\beta-3} = \beta - 4$ and $a_{\beta-2} = 2$ with value $\frac{\beta-4}{\beta-3} + \frac{2}{\beta-2} = 1 + \frac{\beta-4}{(\beta-2)(\beta-3)} < \frac{\beta}{\beta-1}$. This value is $\geqslant \frac{\beta+1}{\beta}$ for $\beta \geqslant 6$, so that for $\beta \geqslant \max\{\alpha+3, 6\}$ we have that $\mathcal{S}_\alpha(\beta-1) \geqslant \frac{\beta+1}{\beta} > \mathcal{S}_\alpha(\beta)$ holds. This gives the following characterisation.

$$\begin{array}{llllllllllll}
\alpha = 2 & : & \mathcal{S}_2(2) & < & \mathcal{S}_2(3) & = & \mathcal{S}_2(4) & < & \mathcal{S}_2(5) & > & \mathcal{S}_2(6) & > \cdots \\
\alpha \geqslant 3, \alpha \text{ is even} & : & \mathcal{S}_\alpha(\alpha) & < & \mathcal{S}_\alpha(\alpha+1) & = & \mathcal{S}_\alpha(\alpha+2) & > & \mathcal{S}_\alpha(\alpha+3) & > & \mathcal{S}_\alpha(\alpha+4) & > \cdots \\
\alpha \geqslant 3, \alpha \text{ is odd} & : & \mathcal{S}_\alpha(\alpha) & < & \mathcal{S}_\alpha(\alpha+1) & < & \mathcal{S}_\alpha(\alpha+2) & > & \mathcal{S}_\alpha(\alpha+3) & > & \mathcal{S}_\alpha(\alpha+4) & > \cdots
\end{array}$$

4) For $\beta \geqslant \alpha + 2$ we have $\mathcal{S}_\alpha(\beta) \geqslant \mathcal{S}_{(\beta-2)}(\beta) \geqslant \frac{\beta+1}{\beta} - \frac{2}{(\beta-2)(\beta-1)\beta}$, so that an asymptotic characterisation is given by $\mathcal{S}_\alpha(\beta) = 1 + 1/\beta - O(1/\beta^3)$.

5) For α is even we have $\mathcal{S}_\alpha(\alpha+2) = \mathcal{S}_\alpha(\alpha+1)$.

6) With the exception of $\alpha = 2$, the maximum of $\mathcal{S}_\alpha(\beta)$ for a given α is achieved by $\beta = \alpha + 2$.

7) From the definition we directly have $\mathcal{S}_\alpha(\beta) \geqslant \mathcal{S}_{(\alpha+1)}(\beta)$.

8) For $\beta \geqslant 2\alpha$ we have $\mathcal{S}_\alpha(\beta) = \mathcal{S}_{(\alpha+1)}(\beta)$, since $\frac{1}{\alpha}$ is dominated by $\frac{1}{2\alpha}$.

---

[2]For $\beta = \alpha, \alpha+1$ the value of $\lambda_\alpha(\beta)$ is apparent, for the values $\beta = \alpha+2, \alpha+3$ it follows from the following

$$\begin{aligned}
\lambda_\alpha(\alpha+2) &= \operatorname{lcm}\{\alpha, \alpha+1, \alpha+2\} = \operatorname{lcm}\{\alpha, (\alpha+1)(\alpha+2)\} = \frac{\alpha(\alpha+1)(\alpha+2)}{\gcd\{\alpha, \alpha^2+3\alpha+2\}} = \frac{\alpha(\alpha+1)(\alpha+2)}{\gcd\{\alpha, 2\}} \\
\lambda_\alpha(\alpha+3) &= \operatorname{lcm}\{\alpha, \alpha+1, \alpha+2, \alpha+3\} = \operatorname{lcm}\{\alpha, \alpha+1, (\alpha+2)(\alpha+3)\} = \\
&= \frac{\alpha(\alpha+1)(\alpha+2)(\alpha+3)}{\gcd\{\alpha, \alpha^2+5\alpha+6\} \times \gcd\{\alpha+1, \alpha^2+5\alpha+6\}} = \frac{\alpha(\alpha+1)(\alpha+2)(\alpha+3)}{\gcd\{\alpha, 6\} \times \gcd\{\alpha+1, 2\}} = \frac{\alpha(\alpha+1)(\alpha+2)(\alpha+3)}{2\gcd\{\alpha, 3\}}
\end{aligned}$$

$\alpha\backslash\beta$	2	3	4	5	6	7	8	9	10
2	1.00000	1.16667	1.16667	1.18333	1.15000	1.13810	1.12024	1.11071	1.09960
3	–	1.00000	1.16667	1.18333	1.15000	1.13810	1.12024	1.11071	1.09960
4	–	–	1.00000	1.15000	1.15000	1.13810	1.12024	1.11071	1.09960
5	–	–	–	1.00000	1.13333	1.13810	1.12024	1.11071	1.09960
6	–	–	–	–	1.00000	1.11905	1.11905	1.10913	1.09960
7	–	–	–	–	–	1.00000	1.10714	1.10913	1.09841
8	–	–	–	–	–	–	1.00000	1.09722	1.09722
9	–	–	–	–	–	–	–	1.00000	1.08889
10	–	–	–	–	–	–	–	–	1.00000

Table B.3. Solution values of $\mathcal{S}_\alpha(\beta)$

β	$\lambda_\alpha(\beta)$	$\mathcal{S}_\alpha(\beta)$	$\mathbf{a}^*$
α	α	1	$[\alpha]$
$\alpha+1$	$\alpha(\alpha+1)$	$\frac{\alpha+2}{\alpha+1}-\frac{1}{\alpha(\alpha+1)}$	$[\alpha-1,2]$
$\alpha+2$	$\frac{\alpha(\alpha+1)(\alpha+2)}{\gcd\{\alpha,2\}}$	$\frac{\alpha+3}{\alpha+2}-\frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$	$[\frac{\alpha-\gcd\{\alpha,2\}}{2},\gcd\{\alpha,2\},\frac{\alpha+4-\gcd\{\alpha,2\}}{2}]$

Table B.4. Solution of $\mathcal{S}_\alpha(\beta)$ for $\beta \leqslant \alpha+2$

α	$\mathcal{S}_\alpha(\beta)$	value	$\mathbf{a}^*$	Bound (B25)
2	71/60	1.18333	[0, 1, 1, 3]	71/60
3	69/60	1.15000	[0, 1, 2, 3]	69/60
4	478/420	1.13810	[0, 2, 1, 4]	479/420
5	941/840	1.12024	[2, 1, 3, 1]	944/840
6	559/504	1.10913	[0, 3, 1, 5]	559/504
7	2768/2520	1.09841	[1, 0, 5, 4]	2771/2520
8	4319/3960	1.09066	[1, 1, 4, 5]	4319/3960
9	2143/1980	1.08232	[1, 1, 5, 5]	2144/1980
10	9238/8580	1.07669	[1, 2, 4, 6]	9239/8580

Table B.5. Solutions of $\mathcal{S}_\alpha(\alpha+3)$

B.6 Instances for FFD

The optimum solution of (B24) enables us to derive instances that provide lower bounds for the asymptotic ratio of FFD. For given values of α and β, let $\mathbf{a}^*$ be an optimal solution to (B24) with value $\mathcal{S}_\alpha(\beta)$. Now define the list $\mathcal{L}^*$ as follows

$$\text{List } \mathcal{L}^* = \{ \underbrace{\tfrac{1}{\alpha},\ldots,\tfrac{1}{\alpha}}_{a^*_\alpha}, \;\cdots,\; \underbrace{\tfrac{1}{i},\ldots,\tfrac{1}{i}}_{a^*_i}, \;\cdots,\; \underbrace{\tfrac{1}{\beta},\ldots,\tfrac{1}{\beta}}_{a^*_\beta} \} \text{ on bins of size } \mathcal{S}_\alpha(\beta). \tag{B26}$$

Note that by virtue of its definition, this list can be packed into exactly one bin with zero waste. This list further provides us with a feasible solution to the FFD ratio-problem (B12), and thus with a lower bound for the asymptotic ratio of FFD. To see this, scale the size of the items to the bin size; the size of the smallest item is then given by $x = \frac{1}{\beta}/\mathcal{S}_\alpha(\beta) \in \langle\frac{1}{\beta+1}, \frac{1}{\beta}]$ and for all others by $\frac{1}{i}/\mathcal{S}_\alpha(\beta) \in \langle\frac{1-x}{i}, \frac{1}{i}]$.

We define λ as the smallest scalar of $\{a^*_i/i\}$, $\alpha \leqslant i \leqslant \beta$, the list $\mathcal{L}_0$ as λ copies of $\mathcal{L}^*$; $\mathcal{L}_0 = \lambda\mathcal{L}^*$, and μ as the smallest scalar of $\{a^*_i/i\}$, $\alpha \leqslant i \leqslant \beta - 1$. To avoid confusion, we state again that the bin size for the remainder of this section is assumed to be $\mathcal{S}_\alpha(\beta)$.

B29 Lemma $\forall k \in \mathbb{N}^+$ $\mathit{FFD}(k\mathcal{L}_0)/\mathit{OPT}(k\mathcal{L}_0) = \mathcal{S}_\alpha(\beta)$.

Proof. *The first bin in the FFD-packing contains exactly α items of size $1/\alpha$. No more items will fit since the smallest item is $1/\beta$ and $1 + 1/\beta$ exceeds the bin size, use (B25). Since $\alpha|\lambda a^*_\alpha$ there are exactly $k\lambda a^*_\alpha/\alpha$ of these bins. Repeating this argument we see that $\mathit{FFD} = \sum_{i=\alpha}^{\beta} k\lambda a^*_i/i = k\lambda\mathcal{S}_\alpha(\beta)$. The list $k\mathcal{L}_0$ can be packed into exactly $k\lambda$ bins with no waste, since it is a 'multiple' of $\mathcal{L}^*$, and thus $\mathit{OPT}(k\mathcal{L}_0) = \mathit{CSP}_R(k\mathcal{L}_0) = \mathit{Mat}(k\mathcal{L}_0) = k\lambda$. This proves the lemma* □

Note that this lemma implies that the value of (B24) is a lower bound for the asymptotic ratio of FFD for lists on $\langle\frac{1}{\beta+1}, \frac{1}{\alpha}]$. We now extend this lemma to give lower bounds for worst-case instances.

B30 Lemma *If* $\gcd\{\mu a^*_\beta, \beta\} = 1$ *then* $\forall k\ \exists\{\mathcal{L} \mid \mathit{OPT}(\mathcal{L}) \geqslant k\}$ $\mathit{FFD}(\mathcal{L}) = 1 - \frac{1}{\beta} + \mathcal{S}_\alpha(\beta)\mathit{OPT}(\mathcal{L})$

Proof. *There exists a $\tilde{k}$ such that ($\tilde{k}\mu a^*_\beta \equiv 1 \bmod \beta$) if and only if $\gcd\{\mu a^*_\beta, \beta\} = 1$ (a well-known result in elementary number theory). We now create a list $\mathcal{L}_1$ by taking $\tilde{k}\mu$ copies of the list $\mathcal{L}^*$. This list has the following characteristics:*

$$\left.\begin{aligned} \mathit{FFD}(\mathcal{L}_1) &= \sum_{i=1}^{\beta-1} \tfrac{\tilde{k}\mu a^*_i}{i} + \tfrac{\tilde{k}\mu a^*_\beta - 1}{\beta} + 1 = 1 - \tfrac{1}{\beta} + \mathcal{S}_\alpha(\beta)\tilde{k}\mu \\ \mathit{OPT}(\mathcal{L}_1) &= \mathit{CSP}_R(\mathcal{L}_1) = \mathit{Mat}(\mathcal{L}_1) = \tilde{k}\mu \end{aligned}\right\} \Rightarrow \mathit{FFD}(\mathcal{L}_1) = 1 - \tfrac{1}{\beta} + \mathcal{S}_\alpha(\beta)\mathit{OPT}(\mathcal{L}_1)$$

That the number of bins used by FFD for each of the different item-types is integer follows from the assumptions made on μ and $\tilde{k}$. The number of bins used by the optimal packing follows since $\mathcal{L}_1$ is a 'multiple' of $\mathcal{L}^*$. To prove the lemma, take the list $\mathcal{L} = k\mathcal{L}_0 + \mathcal{L}_1$, which has $\mathit{FFD}(\mathcal{L}) = 1 - \frac{1}{\beta} + (k\lambda + \tilde{k}\mu)\mathcal{S}_\alpha(\beta)$ and $\mathit{OPT}(\mathcal{L}) = k\lambda + \tilde{k}\mu$, and the lemma follows. □

B31 Lemma $\forall k\ \exists\{\mathcal{L} \mid CSP_R(\mathcal{L}) \geqslant k\}\ FFD(\mathcal{L}) = 1 - \frac{S_\alpha(\beta)}{\beta+1} + S_\alpha(\beta)CSP_R(\mathcal{L})$

Proof. *Take $\mathcal{L}_1$ as a list with just one item of size $S_\alpha(\beta)/(\beta+1)$. This list has the following characteristics:*

$$\left.\begin{aligned} FFD(\mathcal{L}_1) =& 1 \\ CSP_R(\mathcal{L}_1) =& Mat(\mathcal{L}_1) = \tfrac{1}{\beta+1} \end{aligned}\right\} \Rightarrow FFD(\mathcal{L}_1) = 1 - \tfrac{S_\alpha(\beta)}{\beta+1} + S_\alpha(\beta)CSP_R(\mathcal{L}_1)$$

Note that by (B25), $S_\alpha(\beta)/(\beta+1) < 1/\beta$ and $1+S_\alpha(\beta)/(\beta+1) > S_\alpha(\beta)$. This implies, adding the list $\mathcal{L}_1$ to copies of $\mathcal{L}_0$, that the [item in] list $\mathcal{L}_1$ ends up in the last bin as a singleton. Now take the list $\mathcal{L} = k\mathcal{L}_0 + \mathcal{L}_1$, which has $FFD(\mathcal{L}) = 1 + k\lambda S_\alpha(\beta)$ and $CSP_R(\mathcal{L}) = \frac{1}{\beta+1} + k\lambda$, and the lemma follows. □

Examples

To illustrate the lemmas we will use lemma B30 and construct the instances that show that bounds (8.80) and (8.81) are asymptotically tight.

$\beta = \alpha + 1$ and α even For this case (see table B.4) we have $S_\alpha(\beta) = \frac{\alpha^2+2\alpha-1}{\alpha(\alpha+1)}$ and $\mathbf{a}^* = [\alpha-1, 2]$. This gives $\lambda = \alpha(\alpha+1)$ and $\mu = \alpha$. There exists a $\tilde{k}$, since $\gcd\{\mu a^*_\beta, \beta\} = \gcd\{2\alpha, \alpha+1\} = 1$, and is given by $\tilde{k} = \frac{1}{2}\alpha$. A balance for the lists $\mathcal{L}_0$ and $\mathcal{L}_1$ is given in table B.6.

$\beta = \alpha + 2$ and α odd For this case (see table B.4) we have $S_\alpha(\beta) = \frac{\alpha(\alpha+1)(\alpha+3)-1}{\alpha(\alpha+1)(\alpha+2)}$ and $\mathbf{a}^* = [\frac{\alpha-1}{2}, 1, \frac{\alpha+3}{2}]$. This gives $\lambda = \frac{\alpha(\alpha+1)(\alpha+2)}{\gcd\{\alpha,2\}}$ and $\mu = \alpha(\alpha+1)$. There exists a $\tilde{k}$, since $\gcd\{\mu a^*_\beta, \beta\} = 1$, and is given by $\tilde{k} = 1$. A balance for the lists $\mathcal{L}_0$ and $\mathcal{L}_1$ is given in table B.7

Comments

- The lists $\mathcal{L}_0$ and $\mathcal{L}_1$ have the following properties (the same holds in OPT).

$$\begin{aligned} FFD(k\mathcal{L}_0) &= kFFD(\mathcal{L}_0) \\ FFD(k\mathcal{L}_0 + \mathcal{L}_1) &= FFD(k\mathcal{L}_0) + FFD(\mathcal{L}_1) \end{aligned}$$

In analogy with matrix-algebra, one could view the lists $\mathcal{L}_0$ and $\mathcal{L}_1$ as the 'homogeneous' and 'particular' solution.

- The second instance in table B.7 gives rise to the parameter d_α that was determined by Xu (see footnote 12 on page 134).

	FFD-bins		$\mathcal{L}_0$	LP-patterns
d_1	α	$\mathbf{0}$	$\alpha(\alpha^2-1)$	$\boldsymbol{\alpha-1}$
d_2	$\mathbf{0}$	$\boldsymbol{\alpha+1}$	$2\alpha(\alpha+1)$	$\mathbf{2}$
	α^2-1	2α		
	$\alpha^2+2\alpha-1$			$\alpha(\alpha+1)$

Homogeneous solution

	FFD-bins			$\mathcal{L}_1$	LP-patterns
d_1	α	$\mathbf{0}$	$\mathbf{0}$	$\frac{1}{2}\alpha^2(\alpha-1)$	$\boldsymbol{\alpha-1}$
d_2	$\mathbf{0}$	$\boldsymbol{\alpha+1}$	$\mathbf{1}$	α^2	$\mathbf{2}$
	$\frac{1}{2}\alpha(\alpha-1)$	$\alpha-1$	1		
	$\frac{1}{2}(\alpha^2+\alpha)$				$\frac{1}{2}\alpha^2$

Particular solution

Table B.6. Balances for $FFD = (1 - \frac{1}{\alpha+1}) + \frac{\alpha^2+2\alpha-1}{\alpha(\alpha+1)} OPT$, α is even !!!
Instance 1: Items of size $d_1 = \alpha + 1$, $d_2 = \alpha$ on bins of size $L = \alpha^2 + 2\alpha - 1$.
An example for $\alpha = 2$ can be found in diagram 8.10 on page 96.
Instance 2: Items of size $d_1 = \frac{1}{\alpha+1} + \varepsilon$, $d_2 = \frac{1}{\alpha+1} - \frac{\alpha-1}{2}\varepsilon$ on bins of size 1.

	FFD-bins			$\mathcal{L}_0$	LP-patterns
d_1	α	$\mathbf{0}$	$\mathbf{0}$	$\frac{1}{2}(\alpha-1)\alpha(\alpha+1)(\alpha+2)$	$\boldsymbol{\frac{1}{2}(\alpha-1)}$
d_2	$\mathbf{0}$	$\boldsymbol{\alpha+1}$	$\mathbf{0}$	$\alpha(\alpha+1)(\alpha+2)$	$\mathbf{1}$
d_3	$\mathbf{0}$	$\mathbf{0}$	$\boldsymbol{\alpha+2}$	$\frac{1}{2}\alpha(\alpha+1)(\alpha+2)(\alpha+3)$	$\boldsymbol{\frac{1}{2}(\alpha+3)}$
	$\frac{1}{2}(\alpha^2-1)(\alpha+2)$	$\alpha(\alpha+2)$	$\frac{1}{2}\alpha(\alpha+1)(\alpha+3)$		
	$\alpha(\alpha+1)(\alpha+3)-1$				$\alpha(\alpha+1)(\alpha+2)$

Homogeneous solution

	FFD-bins				$\mathcal{L}_1$	LP-patterns
d_1	α	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\frac{1}{2}(\alpha-1)\alpha(\alpha+1)$	$\boldsymbol{\frac{1}{2}(\alpha-1)}$
d_2	$\mathbf{0}$	$\boldsymbol{\alpha+1}$	$\mathbf{0}$	$\mathbf{0}$	$\alpha(\alpha+1)$	$\mathbf{1}$
d_3	$\mathbf{0}$	$\mathbf{0}$	$\boldsymbol{\alpha+2}$	$\mathbf{1}$	$\frac{1}{2}\alpha(\alpha+1)(\alpha+3)$	$\boldsymbol{\frac{1}{2}(\alpha+3)}$
	$\frac{1}{2}(\alpha^2-1)$	α	$\frac{1}{2}(\alpha^2+2\alpha-1)$	1		
	$\alpha(\alpha+2)$					$\alpha(\alpha+1)$

Particular solution

Table B.7. Balances for $FFD = (1 - \frac{1}{\alpha+2}) + \frac{\alpha(\alpha+1)(\alpha+3)-1}{\alpha(\alpha+1)(\alpha+2)} OPT$, α is odd !!!
Instance 1: $d_1 = (\alpha+1)(\alpha+2)$, $d_2 = \alpha(\alpha+2)$, $d_3 = \alpha(\alpha+1)$ on bins of size $L = \alpha(\alpha+1)(\alpha+3) - 1$.
An example for $\alpha = 3$ can be found in diagram 8.38 on page 119.
Instance 2: $z = (\alpha+1)^2 + \varepsilon$, $y = \alpha(\alpha+1) + 2\varepsilon$, $x = \alpha^2 + 1 - \varepsilon$ on bins of size $L = (\alpha+1)^3 - 2\alpha$

B.7 Generator for extremal patterns

Suppose we have a set of positive 'sizes' $L \geqslant d_1 \geqslant \cdots \geqslant d_m > 0$ that defines a knapsack polytope, i.e.

$$\mathcal{A} = \left\{ \mathbf{a} \in \mathbb{N}^m \mid \sum_{i=1}^{m} d_i a_i \leqslant L \right\} \tag{B27}$$

This is the set of feasible patterns in a cutting stock or bin-packing problem. We may assume that L and d_i are integers by theorem A1. The set $\mathcal{A}$ can also be characterised by its *extremal patterns*; the set $\mathcal{E}$

$$\mathcal{E} = \left\{ \mathbf{a} \in \mathcal{A} \mid \sum_{i=1}^{m} d_i a_i > L - d \right\}, \tag{B28}$$

where d denotes the smallest size in the set $\{d_i\}$. Normally, when we solve an instance of CSP, the sizes are given and one can (theoretically) enumerate the pattern set $\mathcal{A}$ or $\mathcal{E}$.

Now suppose we are given an extremal pattern set $\mathcal{E}$. How do we determine a generator, i.e. sizes d_i and L, for this set? This is a problem that is frequently encountered in chapter 8, where FFD is analysed. It can be formulated and solved as an integer program.

$$\mathcal{D} = \boxed{\begin{array}{ll} \text{Min} & L \\ \text{st} & \sum_{i=1}^{m} a_{ij} d_i \leqslant L, \ \forall\, \mathbf{a}_j \in \mathcal{E} \\ & \sum_{i=1}^{m} a_{ij} d_i \geqslant L - d + 1, \ \forall\, \mathbf{a}_j \in \mathcal{E} \\ & d_i \geqslant d \ \text{ and } \ d, d_i, L \in \mathbb{N} \end{array}} \tag{B29}$$

This formulation finds a minimal generator (with respect to L). Note that formulation (B29) does not impose any order on the sizes d_i or a definite requirement on which size is the smallest (if we do know that d_m is the smallest item, we can replace d by d_m).

If one is only interested in feasibility, i.e. is a certain set $\mathcal{E}$ extremal, then (B29) can be solved as an LP by relaxing the integer constraints. The LP will give a rational solution which can be scaled to give an integer solution (not necessarily minimal with respect to L).

There is one 'caveat'. Solving program (B29) may yield a $(\mathbf{d}, L)$ which has an extremal set which is a superset for the original extremal set. This is best illustrated by an example. Take $L_0 = 14$ and $\mathbf{d}_0^{\mathsf{T}} = [6, 5, 3]$. The extremal patterns of the corresponding knapsack polytope are given in table B.8. Solving

	1	2	3	4	5	6
a_1	2	1	1	0	0	0
a_2	0	1	0	2	1	0
a_3	0	1	2	1	3	4

Table B.8. Extremal patterns of $\mathcal{A}_0 = \left\{ \mathbf{a} \in \mathbb{N}^3 \mid 6a_1 + 5a_2 + 3a_3 \leqslant 14 \right\}$

the corresponding IP to find a minimal generator yields $L_1 = 9$ and $\mathbf{d}_1^{\mathsf{T}} = [4,3,2]$. The knapsack polytope $\mathcal{A}_1 = \{\mathbf{a} \in \mathbb{N}^3 \mid 4a_1 + 3a_2 + 2a_3 \leqslant 9\}$ has extremal set $\mathcal{E}_1 = \mathcal{E}_0 \cup [0,3,0]$.

In the context of this thesis, in particular the analysis of FFD, it is sometimes easier to take a more direct approach. When we have a certain configuration with sizes, which are implicitly determined by the assumptions made and the FFD-rule, we can derive from this constraints for an IP.

This is not guaranteed to work. There are occasions where one has to incorporate knowledge on the optimal packing. As an example take the configuration in diagram 8.8 on page 92. We could convert this into an integer program as follows;

$$\mathcal{P} = \begin{array}{|llcl|} \hline \text{Min} & L & & \\ \text{st} & 2z \leqslant L & \text{and} & 2z + x \geqslant L + 1 \\ & z + y \leqslant L & \text{and} & z + y + x \geqslant L + 1 \\ & 3x \leqslant L & \text{and} & 4x \geqslant L + 1 \\ & x, y, z, L \in \mathbb{N} & & \\ \hline \end{array} \tag{B30}$$

The first 'column' in $\mathcal{P}$ represents that the items fit into a bin. The second one represents that FFD cannot place the item in the last (singleton) bin in the previous bins. An optimal solution to $\mathcal{P}$ is $(L, z, y, x) = (6, 3, 3, 2)$. These sizes pack into a configuration as shown in 8.8. But as is easily seen FFD produces an optimal packing. Including the constraints $z + 2x \leqslant L$ and $3y \leqslant L$ in $\mathcal{P}$ produces $(L, z, y, x) = (15, 7, 5, 4)$ which is the result which we intended. These constraints represent the minimal LP-solution (see table 8.5, p. 94). This illustrates that one needs to incorporate information on the optimal as well as the heuristic packing, in order to derive actual sizes.

Comments

(1) If the LP-relaxation of (B29) is infeasible for a certain set $\mathcal{E}$, we know that this set cannot represent a knapsack polytope and therefore cannot be derived from a CSP or BPP.

(2) We can easily add constraints to reflect any further requirements on the generator. For instance, if the largest size is $> \frac{1}{2}L$ (corresponding to a 1-item) add $2d_1 \geqslant L + 1$ (assuming that d_1 is the largest size).

(3) A further application is to reduce the generator of an instance of CSP.

For example $\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^2 \mid 7a_1 + 4a_2 \leqslant 15\}$ has extremal set $\mathcal{E} = \{[2,0],[1,2],[0,3]\}$. Solving the IP gives as a minimal generator $\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^2 \mid 3a_1 + 2a_2 \leqslant 7\}$.

Or consider $\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^3 \mid 7a_1 + 6a_2 + 4a_3 \leqslant 15\}$. Solving the IP yields that this set can also be generated as $\mathcal{A} = \{\mathbf{a} \in \mathbb{N}^3 \mid 3a_1 + 3a_2 + 2a_3 \leqslant 7\}$. This last example, in the context of a CSP, means that we can reduce the dimension of the problem, since two of the sizes are equal.

Appendix C

Lemmas for the First-Fit Decreasing Heuristic

C.1 Introduction

In the analysis of the FFD-heuristic we will encounter knapsack-type problems, where the coefficients depend upon two parameters;

$$\mathcal{KP}(x,y) \;=\; \boxed{\begin{array}{ll} \text{Max} & \sum q_i(y)a_i \\ \text{st} & \sum p_i(x)a_i < p_0(x) \\ & a_i \in \mathbb{N} \end{array}} \tag{C1}$$

Throughout this appendix, $p_i(x)$ and $q_i(y)$ will denote linear functions in x and y, respectively. We assume that x is defined on an interval I, such that its closure, $C(I) = [x_0, x_1]$. The objective is to find a value for y that minimises the maximum of $\mathcal{KP}(x,y)$ over the domain of x;

$$\min_{y} \; \max_{x\in I} \; \mathcal{KP}(x,y) \tag{C2}$$

In the following two sections we develop the tools to tackle problems of the form (C2). First we will show that for fixed y the maximisation problem over x decomposes into two ordinary knapsack problems.

$$\max_{x\in I} \mathcal{KP}(x,y) \;=\; \max\{\mathcal{KP}(x_0,y), \mathcal{KP}(x_1,y)\} \tag{C3}$$

In the second section, we will show how, for fixed x, one can solve the minimisation problem over y as a sequence of standard knapsack problems.

C.2 Maximisation problem

For $x \in [x_0, x_1]$ and $p_i(x)$ a linear function in x defines $\mathcal{A}(x)$ as in (C4). Note that for any fixed x, $\mathcal{A}(x)$ is a knapsack polytope.

$$\mathcal{A}(x) \equiv \left\{\mathbf{a} \in \mathbb{N}^m \;\middle|\; \sum_{i=1}^{m} p_i(x)a_i < p_0(x)\right\} \tag{C4}$$

We will show that any set $\mathcal{A}(x)$ can be described in terms of [the elements of] the boundary sets $\mathcal{A}(x_0)$ and $\mathcal{A}(x_1)$.

C1 Lemma $\mathcal{A}(x) \subseteq \mathcal{A}(x_0) \cup \mathcal{A}(x_1)$

Proof. *We can rewrite the constraint in (C4), by grouping terms as follows, where* $c_0(\mathbf{a})$ *and* $c_1(\mathbf{a})$ *are expressions in* $a_1, \ldots, a_m$.

$$c_1(\mathbf{a})\ x < c_0(\mathbf{a}) \tag{C5}$$

Take an element $\mathbf{a}$ *of* $\mathcal{A}(x)$, *which therefore satisfies (C5), and distinguish between the cases.*

(i) $c_1(\mathbf{a}) > 0 \Rightarrow$ *every* $y \leqslant x$ *will satisfy (C5)* $\Rightarrow \forall y \leqslant x \;\; \mathbf{a} \in \mathcal{A}(y)$

(ii) $c_1(\mathbf{a}) = 0 \Rightarrow$ *every* y *will satisfy (C5)* $\Rightarrow \forall y \;\; \mathbf{a} \in \mathcal{A}(y)$

(iii) $c_1(\mathbf{a}) < 0 \Rightarrow$ *every* $y \geqslant x$ *will satisfy (C5)* $\Rightarrow \forall y \geqslant x \;\; \mathbf{a} \in \mathcal{A}(y)$

So, if $\mathbf{a}$ *is an element of* $\mathcal{A}(x)$ *it must also be an element of either* $\mathcal{A}(x_0)$ *or* $\mathcal{A}(x_1)$, *or both. This implies that every element of* $\mathcal{A}$ *is an element of the union of the boundary sets and thus proves the lemma.* □

Since [the elements of] the boundary sets form the building blocks of each set $\mathcal{A}(x)$, we will characterise them further by the following corollary (which is a direct consequence of the proof of lemma C1).

C2 Corollary ***(Characterisation of boundary sets.)***

(i) $\mathbf{a} \in \mathcal{A}(x_0)$ *and* $c_1(\mathbf{a}) \leqslant 0 \;\Rightarrow\; \forall_{x \geqslant x_0} \;\; \mathbf{a} \in \mathcal{A}(x)$

(ii) $\mathbf{a} \in \mathcal{A}(x_0)$ *and* $c_1(\mathbf{a}) > 0 \;\Rightarrow\; \forall_{x < c_0/c_1} \; \mathbf{a} \in \mathcal{A}(x)$ *and* $\forall_{x \geqslant c_0/c_1} \; \mathbf{a} \notin \mathcal{A}(x)$

(iii) $\mathbf{a} \in \mathcal{A}(x_1)$ *and* $c_1(\mathbf{a}) \geqslant 0 \;\Rightarrow\; \forall_{x \leqslant x_1} \;\; \mathbf{a} \in \mathcal{A}(x)$

(iv) $\mathbf{a} \in \mathcal{A}(x_1)$ *and* $c_1(\mathbf{a}) < 0 \;\Rightarrow\; \forall_{x > c_0/c_1} \; \mathbf{a} \in \mathcal{A}(x)$ *and* $\forall_{x \leqslant c_0/c_1} \; \mathbf{a} \notin \mathcal{A}(x)$

C3 Lemma $\bigcup_{x \in \langle x_0, x_1 \rangle} \mathcal{A}(x) = \mathcal{A}(x_0) \cup \mathcal{A}(x_1)$

Proof. *We first prove the following two assertions. For brevity we will use* c_i *to denote* $c_i(\mathbf{a})$.

(1) $\exists \varepsilon > 0 \;\; \mathcal{A}(x_0) \subseteq \mathcal{A}(x_0 + \varepsilon)$

Assume $\mathbf{a} \in \mathcal{A}(x_0)$, *so that* $c_1 x_0 < c_0$, *by (C5). If* $c_1 > 0$ *we have* $x_0 < c_0/c_1$, *so that there is an* $\varepsilon > 0$ *such that* $x_0 < x_0 + \varepsilon < c_0/c_1$, *and therefore by corollary C2.ii that* $\mathbf{a} \in \mathcal{A}(x_0 + \varepsilon)$. *If* $c_1 \leqslant 0$ *then* $\mathbf{a} \in \mathcal{A}(x_0 + \varepsilon)$ *follows directly from corollary C2.i. So, for a suitable small* ε *we have* $\mathbf{a} \in \mathcal{A}(x_0) \Rightarrow \mathbf{a} \in \mathcal{A}(x_0 + \varepsilon)$, *which proves the first assertion.*

(2) $\exists \varepsilon > 0 \;\; \mathcal{A}(x_1) \subseteq \mathcal{A}(x_1 - \varepsilon)$

Assume $\mathbf{a} \in \mathcal{A}(x_1)$, *so that* $c_1 x_1 < c_0$, *by (C5). If* $c_1 < 0$ *we have* $x_1 > c_0/c_1$, *so that there is an* $\varepsilon > 0$ *such that* $c_0/c_1 < x_1 - \varepsilon < x_1$, *and therefore by corollary C2.iv that* $\mathbf{a} \in \mathcal{A}(x_1 - \varepsilon)$. *If* $c_1 \geqslant 0$ *then* $\mathbf{a} \in \mathcal{A}(x_1 - \varepsilon)$ *follows directly from corollary C2.iii. So, for a suitable small* ε *we have* $\mathbf{a} \in \mathcal{A}(x_1) \Rightarrow \mathbf{a} \in \mathcal{A}(x_1 - \varepsilon)$, *which proves the second assertion.*

Assertions 1 and 2 give $\mathcal{A}(x_0) \cup \mathcal{A}(x_1) \subseteq \bigcup_x \mathcal{A}(x)$. *Combining this with lemma C1 proves the lemma.* □

Now consider the following decision problem, where $\mathcal{A}(x)$ is a set of the form (C4).

$$\mathcal{D}(x) = \boxed{\begin{array}{ll} \text{Max} & f(\mathbf{a}) \\ \text{st} & \mathbf{a} \in \mathcal{A}(x) \end{array}} \tag{C6}$$

The following corollary now follows directly from lemma C3.

C4 Corollary $C(I) = [x_0, x_1] \;\Rightarrow\; \max\limits_{x \in I} \mathcal{D}(x) = \max\{\mathcal{D}(x_0), \mathcal{D}(x_1)\}$

After solving $\mathcal{D}(x)$ one might want to perform a sensitivity analysis on the optimal solution. That is, what is the range of x, such that the optimal solution remains optimal or feasible. For this the following corollary is used, which follows from corollary C2.

C5 Corollary ***(Feasibility range of a pattern)*** *The range of* x *for which a pattern* $\mathbf{a}$ *is feasible, i.e.* $\mathbf{a} \in \mathcal{A}(x)$ *is given by;*

(i) $c_1(\mathbf{a}) > 0 \Rightarrow \mathbf{a}$ *feasible for* $x < c_0(\mathbf{a})/c_1(\mathbf{a})$,

(ii) $c_1(\mathbf{a}) = 0 \Rightarrow \mathbf{a}$ *feasible for all* x,

(iii) $c_1(\mathbf{a}) < 0 \Rightarrow \mathbf{a}$ *feasible for* $x > c_0(\mathbf{a})/c_1(\mathbf{a})$.

From corollary C2 it follows that the only x-values on $[x_0, x_1]$ at which changes in $\mathcal{A}(x)$ can occur are the elements of the following sets.

$$\begin{aligned} \mathcal{S}_0 &\equiv \{c_0(\mathbf{a})/c_1(\mathbf{a}) \mid \mathbf{a} \in \mathcal{A}(x_0), c_1(\mathbf{a}) > 0\} : \text{ we lose patterns from } \mathcal{A}(x_0) \\ \mathcal{S}_1 &\equiv \{c_0(\mathbf{a})/c_1(\mathbf{a}) \mid \mathbf{a} \in \mathcal{A}(x_1), c_1(\mathbf{a}) < 0\} : \text{ we gain patterns from } \mathcal{A}(x_1) \end{aligned} \tag{C7}$$

Once we have determined $\mathcal{S}_0$ and $\mathcal{S}_1$, we can construct a range diagram as shown in diagram C.1. The sets $\mathcal{S}_0$ and $\mathcal{S}_1$ can be determined by either one of the following two methods:

enumeration: simply enumerate all patterns $\mathbf{a} \in \mathcal{A}(x_0)$ and determine the values $c_0(\mathbf{a})/c_1(\mathbf{a})$ which are in the interval $[x_0, x_1]$. This gives gives the set $\mathcal{S}_0$. The same procedure can be applied to determine the set $\mathcal{S}_1$ by using $\mathcal{A}(x_1)$ instead of $\mathcal{A}(x_0)$.

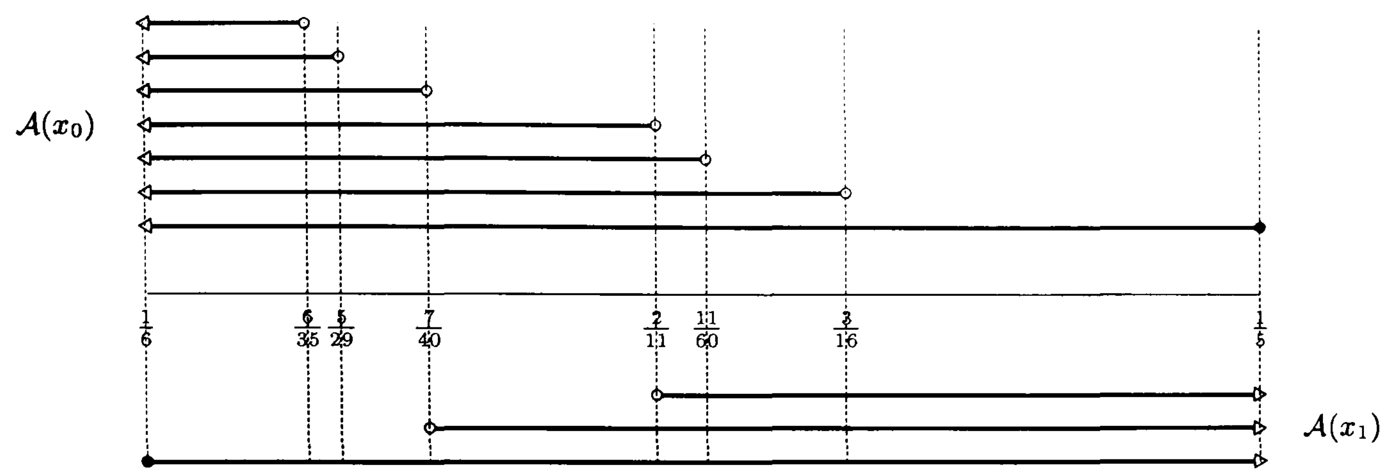


Diagram C.1. Range diagram
This range diagram originates from the following sets, where $x \in \langle \frac{1}{6}, \frac{1}{5} \rangle$.

$$\mathcal{A}(x) = \left\{ \mathbf{a} \in \mathbb{N}^8 \,\middle|\, \tfrac{1}{2}a_1 + \tfrac{1-x}{2}a_2 + \tfrac{1}{3}a_3 + \tfrac{1-x}{3}a_4 + \tfrac{1}{4}a_5 + \tfrac{1-x}{4}a_6 + \tfrac{1}{5}a_7 + xa_8 < 1 \right\}$$

The upper half of the diagram corresponds to the [elements of the] set $\mathcal{A}(x_0)$ and the lower half to the set $\mathcal{A}(x_1)$. The arrows indicate feasibility ranges for sets of patterns. For instance, the top arrow indicates that there are patterns in $\mathcal{A}(x_0)$ which are elements of $\mathcal{A}(x)$ only for $x < 6/35$. Similarly, there are patterns in $\mathcal{A}(x_1)$ which are elements of $\mathcal{A}(x)$ only for $x > 2/11$.

knapsack problem: The elements of $\mathcal{S}_0$ are determined by the patterns $\mathbf{a}$, which satisfy $\mathbf{a} \in \mathcal{A}(x_0)$ and $\mathbf{a} \notin \mathcal{A}(x_1)$. Recall that $\mathbf{a} \in \mathcal{A}(x_0)$ implies $c_1 x_0 < c_0$ and $\sum p_i(x_0)a_i < p_0(x_0)$. Further, $\mathbf{a} \notin \mathcal{A}(x_1)$ implies $c_1 x_1 \geqslant c_0$ and $\sum p_i(x_1)a_i \geqslant p_0(x_1)$. Now consider the following knapsack problem;

$$\mathcal{KP}(x_0, x_1) = \boxed{\begin{array}{ll} \text{Min} & \sum p_i(x_0)a_i - p_0(x_0) \\ \text{st} & \sum p_i(x_1)a_i \geqslant p_0(x_1) \\ & a_i \in \mathbb{N} \end{array}} \tag{C8}$$

This problem is feasible if $\mathcal{A}(x_1)$ is non-void. Denote by z^* the solution value of, and by $\mathbf{a}^*$ an optimal solution to $\mathcal{KP}(x_0, x_1)$. Distinguish between the following two cases;

(i) $z^* \geqslant 0 \Rightarrow \sum p_i(x_0)a_i \geqslant p_0(x_0)$ and $c_1 x_0 \geqslant c_0$,

so the following holds; $\forall \mathbf{a} \notin \mathcal{A}(x_1) \Rightarrow \mathbf{a} \notin \mathcal{A}(x_0)$ which implies $\mathcal{A}(x_0) \subseteq \mathcal{A}(x_1)$.

(ii) $z^* < 0 \Rightarrow \sum p_i(x_0)a_i < p_0(x_0)$ and $c_1 x_0 < c_0$,

so the following holds; $\exists \mathbf{a} \in \mathcal{A}(x_0)$ such that $\mathbf{a} \notin \mathcal{A}(x_1)$. This implies $\mathcal{A}(x_0) \not\subseteq \mathcal{A}(x_1)$.

Therefore, solving $\mathcal{KP}(x_0, x_1)$ either yields that $\mathcal{A}(x_0) \subseteq \mathcal{A}(x_1)$, in which case $\mathcal{S}_0 = \emptyset$, or an element of $\mathcal{S}_0$, namely $c_0(\mathbf{a}^*)/c_1(\mathbf{a}^*)$. In the first case we can stop and in the second we can divide the interval and repeat the process. The same procedure can be applied to determine the set $\mathcal{S}_1$, by interchanging x_0 and x_1 in the above.

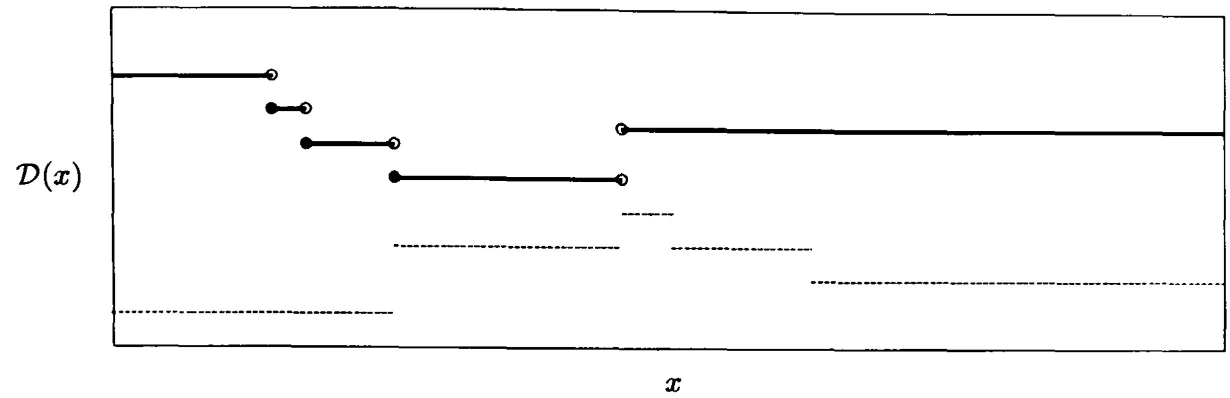


Diagram C.2. $\mathcal{D}$ as intersection of step-functions.

The range diagram indicates the structure of $\mathcal{D}(x)$ as a function of x. If we enumerate and order the elements $\mathbf{a}$ of $\mathcal{A}(x_0)$ to their pattern values $f(\mathbf{a})$ as we have suggestively done in the range diagram and take the maximum of the pattern values we see that we end up with a step function. Repeating the process for $\mathcal{A}(x_1)$ we end up with a diagram similar to C.2. It shows that $\mathcal{D}(x)$ can be interpreted as taking the maximum of two [intersecting] step functions. Moreover, this property does not depend on the precise nature of the objective function f; it solely depends on the structure of $\mathcal{A}(x)$. From the U-shaped structure we can see that $\mathcal{D}(x)$ is maximised at one of the boundary values of x.

C.3 Minimisation problem

Consider the following decision problem, where $q_i(y)$ is a linear function in y.

$$\mathcal{D}(y) = \boxed{\begin{array}{ll} \text{Max} & \sum q_i(y) a_i \\ \text{st} & \mathbf{a} \in \mathcal{A} \end{array}} \tag{C9}$$

The objective is to find a value for y that minimises $\mathcal{D}(y)$. Rewriting the objective function in (C9), by grouping terms we see that each element $\mathbf{a}$ of $\mathcal{A}$ defines a linear function in the variable y.

$$c_0(\mathbf{a}) + c_1(\mathbf{a})y \tag{C10}$$

From this it follows that $\mathcal{D}(y)$, as the maximum over a set of linear (convex) functions, is a convex and piecewise linear function in y. Diagram C.3 depicts this situation. The dashed lines represent the pattern values (C10) as a function of y and the solid line represents the maximum of all pattern values for a specific value of y. Note that the structure of $\mathcal{D}(y)$, as a convex, piecewise linear function does not depend upon the precise structure of $\mathcal{A}$. The objective is to choose a value of y that minimises $\mathcal{D}(y)$, or conversely minimises the maximum pattern-value, hence the MinMax label. Note that $\mathcal{D}(y)$ either has a unique minimum or a closed interval on which the minimum is achieved. To determine $\mathcal{D}(y)$ we can make use of the following lemma.

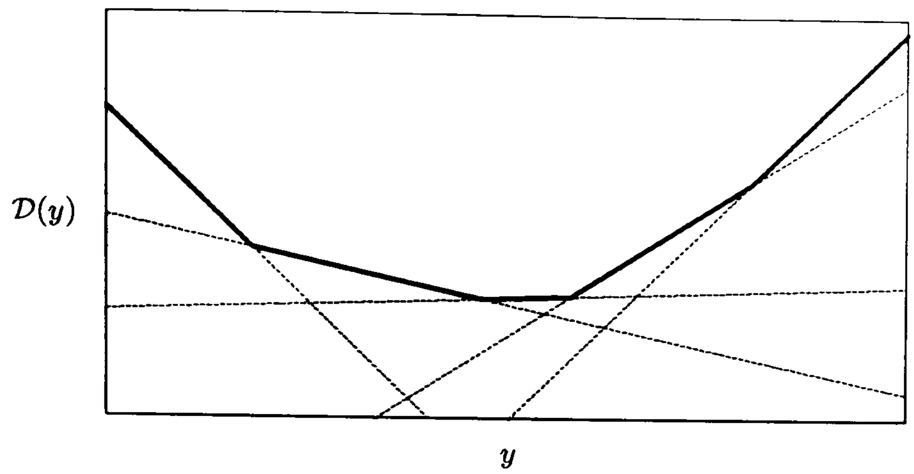


Diagram C.3. MinMax problem.

C6 Lemma *If* $\mathbf{a}$ *is optimal for* $\mathcal{D}(y_1)$ *and* $\mathcal{D}(y_2)$ *then* $\mathbf{a}$ *is optimal for all* $y \in [y_1, y_2]$

Proof. *from the optimality of* $\mathbf{a}$ *we have for any* $\tilde{\mathbf{a}} \in \mathcal{A}$

$$c_0(\tilde{\mathbf{a}}) + c_1(\tilde{\mathbf{a}})y_1 \leqslant c_0(\mathbf{a}) + c_1(\mathbf{a})y_1 \tag{C11}$$

$$c_0(\tilde{\mathbf{a}}) + c_1(\tilde{\mathbf{a}})y_2 \leqslant c_0(\mathbf{a}) + c_1(\mathbf{a})y_2 \tag{C12}$$

Taking the convex sum, with $0 < \lambda < 1$, *gives*

$$c_0(\tilde{\mathbf{a}}) + c_1(\tilde{\mathbf{a}})(\lambda y_1 + (1-\lambda)y_2) \leqslant c_0(\mathbf{a}) + c_1(\mathbf{a})(\lambda y_1 + (1-\lambda)y_2) \tag{C13}$$

Substituting $\lambda = \frac{y-y_2}{y_1-y_2}$ *shows that* $\mathbf{a}$ *is optimal for* $y \in \langle y_1, y_2 \rangle$, *from which the lemma follows.* □

To determine $\mathcal{D}(y)$ on the interval $[y_0, y_1]$ we first solve $\mathcal{D}(y_0)$ and $\mathcal{D}(y_1)$. If $\mathbf{a}_0^*$ and $\mathbf{a}_1^*$ are the optimal solutions to $\mathcal{D}(y_0)$ and $\mathcal{D}(y_1)$, respectively and $\mathbf{a}_0^*$ is optimal for $\mathcal{D}(y_1)$ or conversely $\mathbf{a}_1^*$ is optimal for $\mathcal{D}(y_0)$ then we are done. If not, we calculate the y-value, y_2 at which the two lines corresponding to $\mathbf{a}_0^*$ and $\mathbf{a}_1^*$ intersect and repeat the process for the intervals $[y_0, y_2]$ and $[y_2, y_1]$. Since each evaluation determines a line segment, this process takes at most $O(n)$ evaluations of the IP-problem $\mathcal{D}(y)$, where n is the number of line segments in the function $\mathcal{D}(y)$. Note that this will give us the entire function $\mathcal{D}(y)$. If we are only interested in the minimum, one can do a straightforward binary search, making use of lemma C6.

Appendix D

Weighting functions for the First-Fit Decreasing Heuristic

In this appendix we have collected the derivation of the weighting functions and associated programs for the various subcases in the analysis of FFD. The structure is roughly the same for each case.

Ratio

1. Introduce a family of recurrent weighting functions, which depends upon a parameter V.
2. Formulate the ratio problem $\mathcal{R}(V)$.
3. Determine the ratio as a function of V and find the value of V that minimises $\mathcal{R}(V)$. This determines the (minimal) weighting function.

Constant

1. Perturbate the (minimal) weighting function to find a *stronger* weighting function with the same ratio.
2. Enumerate all patterns that correspond to transition bins with a weight strictly less than 1. Now set up a set-packing problem (SPP) that gives an upper bound for the constant.

Problem instance

1. Set up a balance between the FFD-patterns and the patterns that achieve the maximum ratio. This determines the usage of the patterns in the FFD-configuration and the optimal LP-solution.
2. Solve a *generator* problem to find a set of [item] sizes that will generate the pattern set. This determines the actual sizes in the list.

D.1 $x \in \langle \frac{1}{5}, \frac{1}{4}]$

D.1.1 $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1}{3}, \frac{1-x}{2}]$

We use the following weights and weighting function, with $V \in [\frac{1}{2}, \frac{5}{8}]$ to be determined.

$$W(z) = V,\ W(v) = 1 - V,\ W(w) = \tfrac{1}{2} \text{ and } W(s) = \begin{cases} \frac{3}{8}, & s \in \langle \frac{1}{3}, v \rangle \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1}{3}] \\ \frac{1}{4}, & s \in [x, \frac{1-x}{3}] \end{cases} \tag{D1}$$

The choice of range for V ensures that the weights are non-decreasing. As can be seen from diagram 8.18, these weights further ensure that all recurrent bins have a bin weight of at least 1. An upper bound for the asymptotic ratio now follows from the following knapsack problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + xa_4 \leqslant 1 \\ & x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D2}$$

Note that we have deleted items in $\langle \frac{1}{3}, v \rangle$ from the formulation. This can be done because in the above formulation any such item is dominated, since $1-V \geqslant \frac{3}{8}$. Note further that we can delete item w; since $w > 2x$ any such item is dominated by item x. The constraint can be rewritten using the following equivalence.

$$\boxed{\begin{array}{c} \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + xa_4 \leqslant 1 \\ \text{and} \\ x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \end{array}} \cong \boxed{\begin{array}{c} 6a_1 + 4a_2 + 3a_3 + 3a_4 \leqslant 11 \quad (x = \frac{1}{4}) \\ \text{or} \\ 15a_1 + 10a_2 + 8a_3 + 6a_4 \leqslant 29 \quad (x = \frac{1}{5}) \end{array}} \tag{D3}$$

The constraint for $(x = \frac{1}{4})$ is implied by the constraint for $(x = \frac{1}{5})$, so that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & 15a_1 + 10a_2 + 8a_3 + 6a_4 \leqslant 29 \\ & a_i \in \mathbb{N} \end{array}} \tag{D4}$$

The extremal patterns of program (D4) are enumerated in table D.1. A generator for this set is given by the constraint in program (D4). This directly gives values for z, v, y and x as $(z, v, y, x) = \frac{1}{29}(15, 10, 8, 6)$. To determine the minimising value of V we graph the table as is done in diagram D.1. Intersecting the bold lines, corresponding to patterns 2 and 4 gives $V + \frac{7}{12} = -2V + \frac{7}{3}$, so that the minimum of program (D2) is attained for a value of $V = \frac{7}{12}$, which yields a ratio of $\frac{7}{6}$. Substituting $V = \frac{7}{12}$ in (D1) gives $W(z) = \frac{7}{12}$ and $W(v) = \frac{5}{12}$. The balance equation using only sizes z, v, y and x is given in table D.2. From the formulation of program (D2) we see that we can increase the weight of all items in $\langle \frac{1}{3}, v \rangle$ to $1 - V = \frac{5}{12}$, without increasing its solution value.

	a_1	a_2	a_3	a_4	value	V= 7/12
1	1	1	0	0	1	12/12
2	1	0	1	1	V + 7/12	→ 14/12 ←
3	1	0	0	2	V + 1/2	13/12
4	0	2	1	0	-2V + 7/3	→ 14/12 ←
5	0	2	0	1	-2V + 9/4	13/12
6	0	1	2	0	-V + 5/3	13/12
7	0	1	1	1	-V + 19/12	12/12
8	0	1	0	3	-V + 7/4	→ 14/12 ←
9	0	0	3	0	1	12/12
10	0	0	2	2	7/6	→ 14/12 ←
11	0	0	1	3	13/12	13/12
12	0	0	0	4	1	12/12

Table D.1. Extremal patterns of program (D4)
Generator: $\mathcal{A} = \{15a_1 + 10a_2 + 8a_3 + 6a_4 \leqslant 29\}$

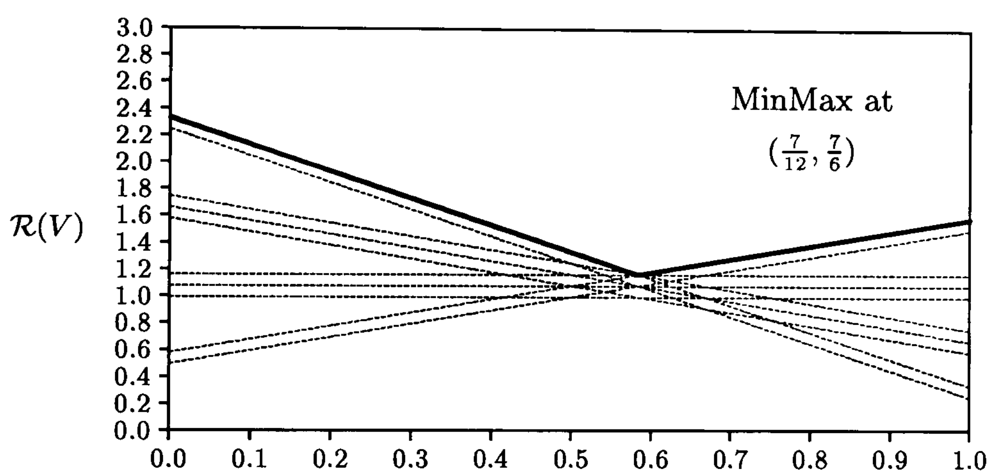


Diagram D.1. Pattern values for program (D4).

	FFD-bins					LP-patterns			
$z = \frac{15}{29}$	1	0	0	0	4k	1	0	0	$u_1^* = \frac{1}{2}$
$v = \frac{10}{29}$	1	0	0	0	4k	0	2	0	$u_2^* = \frac{1}{3}$
$y = \frac{8}{29}$	0	3	0	0	6k+9	1	1	2	$u_3^* = \frac{1}{3}$
$x = \frac{6}{29}$	0	0	4	1	4k+9	1	0	2	$u_4^* = \frac{1}{6}$
	4k	2k+3	k+2	1		4k	2k	$\frac{9}{2}$	
	7k+6					$6k + \frac{9}{2}$			

Table D.2. Balance with $FFD = \frac{3}{4} + \frac{7}{6} CSP_R$.

D.1.2 $x \in \langle\frac{1}{5},\frac{1}{4}\rangle$ and $v \in \langle\frac{1}{2}-x,\frac{1}{3}]$

We start with the weights for the recurrent items, where $V \in [\frac{1}{2},\frac{2}{3}]$ is to be determined.

$$W(z) = V,\ W(v) = 1-V,\ W(y) = \frac{1}{3} \text{ and } W(x) = \frac{1}{4} \tag{D5}$$

The choice of range for V ensures that the weights are non-decreasing. As can be seen from diagram 8.19, these weights further ensure that all recurrent bins have a bin weight of 1. The asymptotic ratio now follows from the following knapsack problem.

$$\mathcal{R}(V) = \begin{array}{|ll|}\hline \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1}{2}-x\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + xa_4 \leqslant 1 \\ & x \in \langle\frac{1}{5},\frac{1}{4}\rangle \text{ and } a_i \in \mathbb{N} \\ \hline\end{array} \tag{D6}$$

The constraint can be rewritten using the following equivalence.

$$\begin{array}{|c|}\hline \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1}{2}-x\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + xa_4 \leqslant 1 \\ \text{and} \\ x \in \langle\frac{1}{5},\frac{1}{4}\rangle \\ \hline\end{array} \cong \begin{array}{|c|}\hline 2a_1 + a_2 + a_3 + a_4 \leqslant 3 \quad (x=\frac{1}{4}) \\ \text{or} \\ 15a_1 + 9a_2 + 8a_3 + 6a_4 \leqslant 29 \quad (x=\frac{1}{5}) \\ \hline\end{array} \tag{D7}$$

The constraint for $\left(x=\frac{1}{4}\right)$ is implied by the constraint for $\left(x=\frac{1}{5}\right)$, so that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \begin{array}{|ll|}\hline \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & 15a_1 + 9a_2 + 8a_3 + 6a_4 \leqslant 29 \\ & a_i \in \mathbb{N} \\ \hline\end{array} \tag{D8}$$

The extremal patterns of program (D8) are listed in table D.3. A generator for this set is given by the constraint in program (D8). This directly gives values for z, v, y and x as $\frac{15}{29}$, $\frac{9}{29}$, $\frac{8}{29}$ and $\frac{6}{29}$. The pattern values are graphed in diagram D.2. Intersecting the bold lines corresponding to patterns 2 and 8 gives $V + \frac{7}{12} = -V + \frac{11}{6}$, so that the minimum of program (D6) is attained for a value of $V = \frac{5}{8}$, which yields a ratio of $\frac{29}{24}$. Substitution of $V = \frac{5}{8}$ in (D5) gives the weights for z, v, y and x as used in (8.34). The balance equation using only the recurrent sizes z, v, y and x is given in table D.4.

D1 Lemma $w + 2y > 1$

Proof. *If there is an item y then $z < 2y$ must hold by the cutting principle. Otherwise we can replace bin (z, v) by bin (y, y, y). Since $z + w > 1$ it follows that $w > 1 - z > 1 - 2y$ holds and the lemma follows.* □

We set $W(y_1) = W(y_2) = \frac{1}{4}$ and $W(w) = \frac{29}{48}$, and need to show that the maximum pattern-weight does not exceed 29/24. Any pattern without an item w has weight of at most 29/24 (follows directly from the ratio problem). Any pattern with an item w can contain at most two additional items, since $w > 2x$. If there is one then the maximum weight is $W(w) + W(w) = 29/24$. If there are two then the maximum weight is $W(y) + W(y) + W(x) = 19/16$. Note that by lemma D1 (w, y, y) does not represent a feasible bin.

	a_1	a_2	a_3	a_4	value	V= 5/8
1	1	1	0	0	1	24/24
2	1	0	1	1	V + 7/12	→ 29/24 ←
3	1	0	0	2	V + 1/2	27/24
4	0	3	0	0	-3V + 3	27/24
5	0	2	1	0	-2V + 7/3	26/24
6	0	2	0	1	-2V + 9/4	24/24
7	0	1	2	0	-V + 5/3	25/24
8	0	1	1	2	-V + 11/6	→ 29/24 ←
9	0	1	0	3	-V + 7/4	27/24
10	0	0	3	0	1	24/24
11	0	0	2	2	7/6	28/24
12	0	0	1	3	13/12	26/24
13	0	0	0	4	1	24/24

Table D.3. Extremal patterns of program (D6)
Generator: $\mathcal{A} = \{15a_1 + 9a_2 + 8a_3 + 6a_4 \leqslant 29\}$

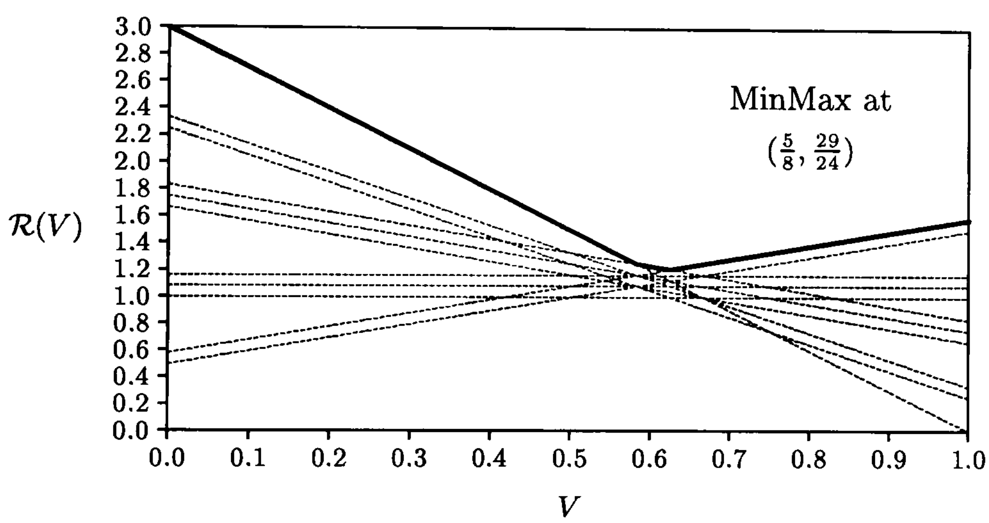


Diagram D.2. Pattern values for program (D6).

	FFD-bins					LP-patterns		
$z = \frac{15}{29}$	**1**	**0**	**0**	**0**	12k+3	**1**	**0**	$u_1^* = \frac{1}{2}$
$v = \frac{9}{29}$	**1**	**0**	**0**	**0**	12k+3	**0**	**1**	$u_2^* = \frac{1}{3}$
$y = \frac{8}{29}$	**0**	**3**	**0**	**0**	24k+6	**1**	**1**	$u_3^* = \frac{1}{3}$
$x = \frac{6}{29}$	**0**	**0**	**4**	**1**	36k+9	**1**	**2**	$u_4^* = \frac{1}{6}$
	12k+3	8k+2	9k+2	1		12k+3	12k+3	
	29k+8					**24k+6**		

Table D.4. Balance with $FFD = \frac{3}{4} + \frac{29}{24} CSP_R$.

D.1.3 $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$

We use the following weights, with $V \in [\frac{1}{2}, \frac{2}{3}]$ to be determined.

$$W(z) = V,\ W(v) = 1 - V,\ W(y) = \frac{1}{3} \text{ and } W(x) = \frac{1}{4} \tag{D9}$$

The choice of range for V ensures that the weights are non-decreasing. As can be seen from diagram 8.20, these weights further ensure that all recurrent bins have a bin weight of 1. The asymptotic ratio now follows from the following knapsack problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{3}a_3 + \frac{1}{4}a_4 \\ \text{st} & (1-v-x)^+ a_1 + va_2 + (\frac{1-x}{3})^+ a_3 + xa_4 \leqslant 1 \\ & a_i \in \mathbb{N},\ v \in \langle \frac{1-x}{3}, \frac{1}{2} - x] \text{ and } x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \end{array}} \tag{D10}$$

We solve program (D10) by enumerating the extremal patterns as done in table D.5. The value for each pattern is depicted in diagram D.3. Intersecting the bold lines corresponding to patterns 2 and 6 gives $V + \frac{7}{12} = 2(1-V) + \frac{1}{2}$, so that the minimum of (D10) is attained for a value of $V = \frac{23}{36}$, which yields a ratio of $\frac{11}{9}$. Substitution of V in (D9) gives the weights as used in (8.36). The balance equation on sizes z, v, y and x is given in table D.6. To find values for the sizes we solve an IP on the extremal patterns. This gives values for z, v, y and x as $\frac{23}{44}$, $\frac{13}{44}$, $\frac{12}{44}$ and $\frac{9}{44}$, respectively.

D.1.4 $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle \frac{1}{4}, \frac{1-x}{3}]$

We use the following weights, where $V \in [\frac{1}{2}, \frac{3}{4}]$ is to be determined.

$$W(z) = V,\ W(v) = 1 - V \text{ and } W(x) = \tfrac{1}{4} \tag{D11}$$

The choice of range for V ensures that the weights are non-decreasing. As can be seen from diagram 8.21 these weights further ensure that all recurrent bins have a bin weight of 1. The asymptotic ratio now follows from the following knapsack problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{4}a_3 \\ \text{st} & (1-v-x)^+ a_1 + va_2 + xa_3 \leqslant 1 \\ & a_i \in \mathbb{N},\ v \in \langle \frac{1}{4}, \frac{1-x}{3}] \text{ and } x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \end{array}} \tag{D12}$$

We solve program (D12) by enumerating the extremal patterns as is done in table D.7. To determine the minimising value of V the pattern values are graphed in diagram D.4. From this diagram we can see that intersecting the bold lines, corresponding to patterns 2 and 3 gives $V + \frac{1}{2} = -3V + \frac{13}{4}$, and a minimising value of $V = \frac{11}{16}$. This gives a value of $\frac{19}{16}$ for the ratio problem (D12). Substitution of V in (D11) now gives the weights $W(z) = \frac{11}{16}$ and $W(v) = \frac{5}{16}$, as used in paragraph 8.8.5. A balance is given in table D.8.

	a_1	a_2	a_3	a_4	value	V=23/36
1	1	1	0	0	1	36/36
2	1	0	1	1	V + 7/12	→ 44/36 ←
3	1	0	0	2	V + 1/2	41/36
4	0	3	0	0	-3V + 3	39/36
5	0	2	1	0	-2V + 7/3	38/36
6	0	2	0	2	-2V + 5/2	→ 44/36 ←
7	0	1	2	0	-V + 5/3	37/36
8	0	1	1	2	-V + 11/6	43/36
9	0	1	0	3	-V + 7/4	40/36
10	0	0	3	0	1	36/36
11	0	0	2	2	7/6	42/36
12	0	0	1	3	13/12	39/36
13	0	0	0	4	1	36/36

Table D.5. Extremal patterns of program (D10)
Generator: $\mathcal{A} = \{23a_1 + 13a_2 + 12a_3 + 9a_4 \leqslant 44\}$

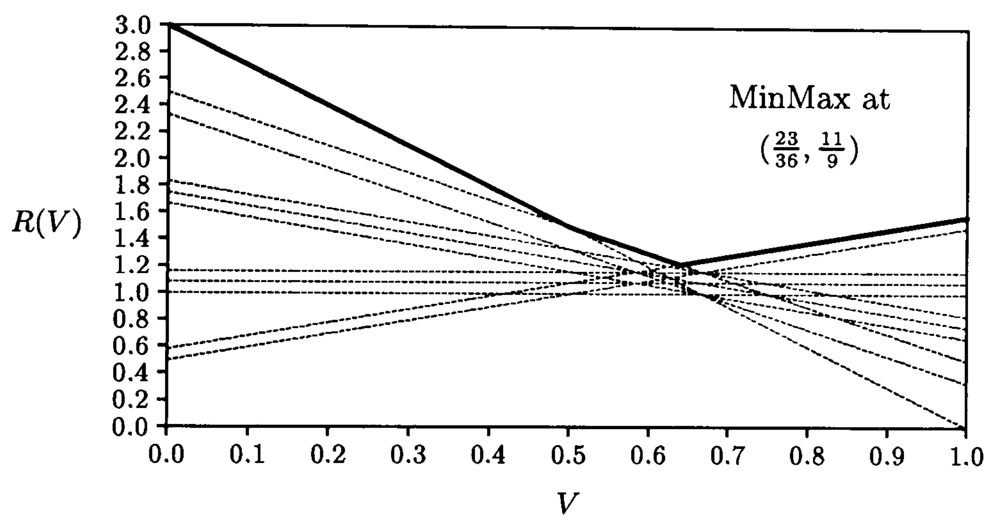


Diagram D.3. Pattern values for program (D10).

	FFD-bins					LP-patterns			
$z = \frac{23}{44}$	**1**	**0**	**0**	**0**	6k+2	**1**	**0**	**0**	$u_1^* = \frac{1}{2}$
$v = \frac{13}{44}$	**1**	**0**	**0**	**0**	6k+2	**0**	**2**	**0**	$u_2^* = \frac{1}{3}$
$y = \frac{12}{44}$	**0**	**3**	**0**	**0**	6k+3	**1**	**0**	**2**	$u_3^* = \frac{1}{3}$
$x = \frac{9}{44}$	**0**	**0**	**4**	**1**	12k+5	**1**	**2**	**2**	$u_4^* = \frac{1}{6}$
	6k+2	2k+1	3k+1	1		6k+2	3k+1	$\frac{1}{2}$	
	11k+5					$\mathbf{9k + \frac{7}{2}}$			

Table D.6. Balance with $FFD = \frac{13}{18} + \frac{11}{9} CSP_R$.

	a_1	a_2	a_3	value	V=11/16
1	1	1	0	1	16/16
2	1	0	2	V + 1/2	→ 19/16 ←
3	0	3	1	-3V + 13/4	→ 19/16 ←
4	0	2	2	-2V + 5/2	18/16
5	0	1	3	-V + 7/4	17/16
6	0	0	4	1	16/16

Table D.7. Extremal patterns of program (D12)
Generator: $\mathcal{A} = \{11a_1 + 5a_2 + 4a_3 \leqslant 19\}$

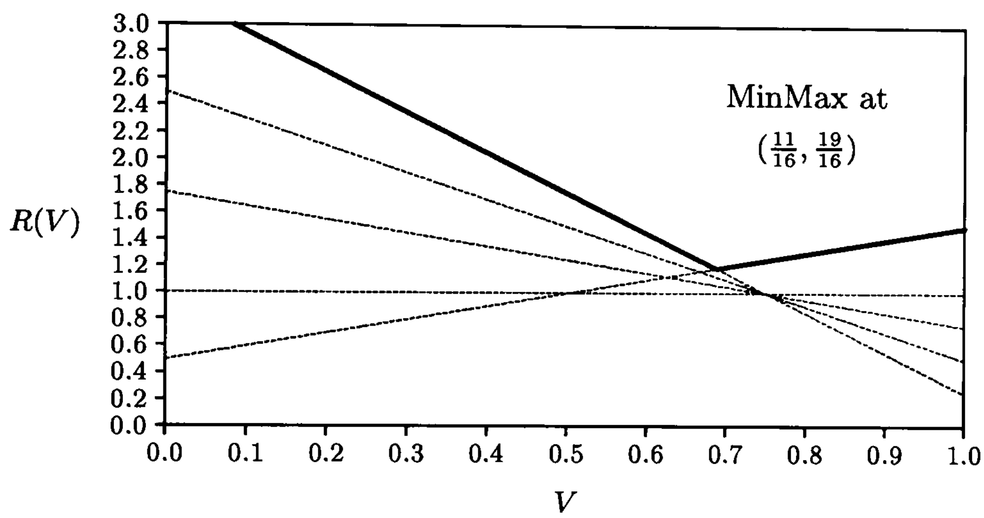


Diagram D.4. Pattern values for program (D12).

	FFD-bins				LP-patterns		
$z = \frac{11}{19}$	**1**	**0**	**0**	12k+9	**1**	**0**	$u_1^* = \frac{1}{2}$
$v = \frac{5}{19}$	**1**	**0**	**0**	12k+9	**0**	**3**	$u_2^* = \frac{1}{4}$
$x = \frac{4}{19}$	**0**	**4**	**1**	28k+21	**2**	**1**	$u_3^* = \frac{1}{4}$
	12k+9	7k+5	1		12k+9	4k+3	
	19k+15				**16k+12**		

Table D.8. Balance for $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$

D.1.5 $x \in \langle \frac{1}{5}, \frac{1}{4} \rangle$ and $v \in \langle x, \frac{1}{4}]$

We use the following weights, where $V \in [\frac{1}{2}, \frac{3}{4}]$ is to be determined.

$$W(z) = V,\ W(v) = 1 - V \text{ and } W(x) = \tfrac{1}{4} \tag{D13}$$

The choice of range for V ensures that the weights are non-decreasing. As can be seen from diagram 8.21 these weights further ensure that all recurrent bins have a bin weight of 1. The asymptotic ratio now follows from the following knapsack problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{1}{4}a_3 \\ \text{st} & (1-v-x)^+ a_1 + va_2 + xa_3 \leqslant 1 \\ & a_i \in \mathbb{N},\ v \in \langle x, \frac{1}{4}]\ \text{and}\ x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \end{array}} \tag{D14}$$

We solve program (D14) by enumerating the extremal patterns as is done in table D.9. To determine the minimising value of V the pattern values are graphed in diagram D.5. Intersecting the bold lines corresponding to patterns 2 and 3 gives $V + \frac{1}{2} = -4V + 4$ and a minimising value of $V = \frac{7}{10}$. This gives a value of $\frac{6}{5}$ for the ratio problem (D14). Substitution of V in (D13) gives the weights $W(z) = \frac{7}{10}$ and $W(v) = \frac{3}{10}$, as used in subsection 8.8.6.

The solution value of the LP associated with the minimal FFD-configuration can be characterised as follows.

$$CSP_R = \begin{cases} \frac{3}{4}n_1 + n_2 + \frac{1}{4}, & n_1 \leqslant 2n_2 \\ n_1 + \frac{1}{2}n_2 + \frac{1}{8}, & n_1 \geqslant 2n_2 + 1 \end{cases} \tag{D15}$$

The actual solutions are shown in table D.10. From this table we can derive the [optimal] dual multipliers for the cases $n_1 \leqslant 2n_2$ and $n_1 \geqslant 2n_2 + 1$; viz. $\mathbf{u}^\mathsf{T} = [\frac{1}{2}, \frac{1}{4}, \frac{1}{4}]$ and $\mathbf{u}^\mathsf{T} = [\frac{3}{4}, \frac{1}{4}, \frac{1}{8}]$. A balance for the case $n_1 = 2n_2$, which gives a minimal configuration, is given in table D.11.

	a_1	a_2	a_3	value	V=7/10
1	1	1	0	1	20/20
2	1	0	2	V + 1/2	→ 24/20 ←
3	0	4	0	-4V + 4	→ 24/20 ←
4	0	3	1	-3V + 13/4	23/20
5	0	2	2	-2V + 5/2	22/20
6	0	1	3	-V + 7/4	21/20
7	0	0	4	1	20/20

Table D.9. Extremal patterns of program (D14)
Generator: $\mathcal{A} = \{14a_1 + 6a_2 + 5a_3 \leqslant 24\}$

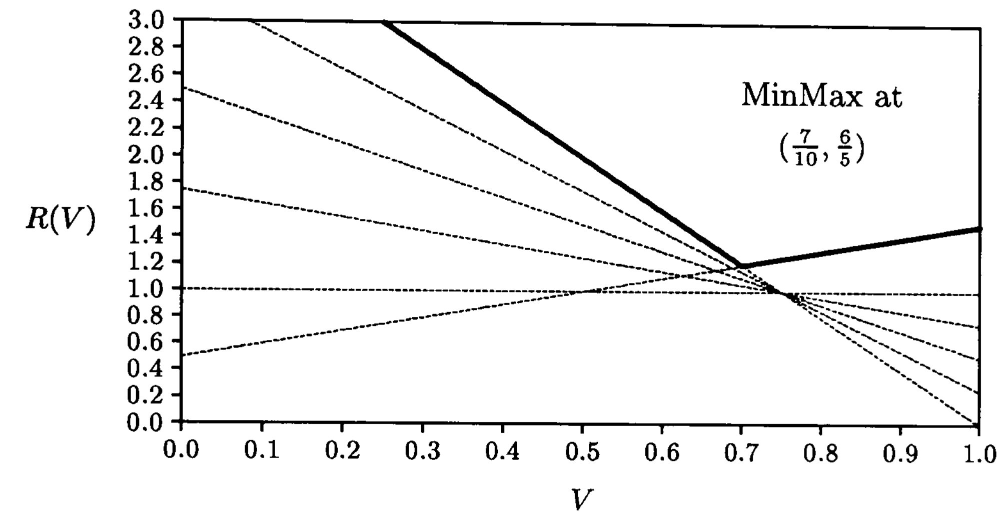


Diagram D.5. Pattern values for program (D14).

	$n_1 \leqslant 2n_2$				$n_1 \geqslant 2n_2 + 1$		
z	**1**	**0**	**0**	n_1	**1**	**0**	**1**
v	**0**	**4**	**0**	n_1	**0**	**4**	**1**
x	**2**	**0**	**4**	$4n_2 + 1$	**2**	**0**	**0**
	n_1	$\frac{1}{4}n_1$	$n_2 - \frac{1}{2}n_1 + \frac{1}{4}$		$2n_2 + \frac{1}{2}$	$\frac{1}{2}n_2 + \frac{1}{8}$	$n_1 - 2n_2 - \frac{1}{2}$
	$\frac{3}{4}n_1 + n_2 + \frac{1}{4}$				$n_1 + \frac{1}{2}n_2 + \frac{1}{8}$		

Table D.10. LP-solutions for the configuration in diagram 8.21

	FFD-bins				LP-patterns			
$z = \frac{14}{24}$	**1**	**0**	**0**	$2k$	**1**	**0**	**0**	$u_1^* = \frac{1}{2}$
$v = \frac{6}{24}$	**1**	**0**	**0**	$2k$	0	**4**	**0**	$u_2^* = \frac{1}{4}$
$x = \frac{5}{24}$	**0**	**4**	**1**	$4k + 1$	**2**	**0**	**4**	$u_3^* = \frac{1}{4}$
	$2k$	k	1		$2k$	$\frac{1}{2}k$	$\frac{1}{4}$	
	3k+1				$\frac{5}{2}k + \frac{1}{4}$			

Table D.11. Balance with $FFD = \frac{7}{10} + \frac{6}{5}CSP_R$

D.1.6 $x \in \left\langle \frac{1}{5}, \frac{1}{4} \right]$ and no 1-item

D.1.6a There is a bin with largest item $y \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right]$

We use the following variant of the generic weighting function.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{1-V}{2}, & s \in \left\langle \frac{1-y}{2}, \frac{1-x}{2} \right] \\ \frac{1}{3}, & s \in \left[y, \frac{1-y}{2} \right] \\ V, & s \in [x, y\rangle \end{cases} \tag{D16}$$

An upper bound for the asymptotic ratio follows from the following knapsack problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + Va_4 \\ \text{st} & \left(\frac{1-x}{2}\right)^+ a_1 + \left(\frac{1-y}{2}\right)^+ a_2 + ya_3 + xa_4 \leqslant 1 \\ & x \in \left\langle \frac{1}{5}, \frac{1}{4} \right],\ y \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D17}$$

The extremal patterns are listed in table D.12. The pattern values are graphed in diagram D.6. Intersecting the bold lines, corresponding to patterns 3 and 11 gives $\frac{7}{6} = 2V + \frac{2}{3}$, so that the minimum value of $\mathcal{R}(V)$ is attained for a value of $V = \frac{1}{4}$, which yields a ratio of $\frac{7}{6}$. Now substitute $V = \frac{1}{4}$ in (D16). Examining the extremal patterns in table D.12 we see that we can increase the weight of the items in $\left\langle \frac{1-y}{2}, \frac{1-x}{2} \right]$ from $\frac{3}{8}$ to $\frac{5}{12}$ without increasing the ratio. This gives the weights as used in (8.43).

D.1.6b There is no bin with largest item $y \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right]$

We use the following weighting function, which can be interpreted as a special case of the previous weighting function by choosing $y = \frac{1}{3}$.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{1-V}{2}, & s \in \left\langle \frac{1}{3}, \frac{1-x}{2} \right] \\ V, & s \in \left[x, \frac{1}{3} \right] \end{cases} \tag{D18}$$

An upper bound for the asymptotic ratio follows from the following knapsack problem. Note that this program is the same as program (D17) with the added constraints $y = \frac{1}{3}$ and $a_3 = 0$,

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + Va_4 \\ \text{st} & \left(\frac{1-x}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + xa_4 \leqslant 1 \\ & x \in \left\langle \frac{1}{5}, \frac{1}{4} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D19}$$

The six extremal patterns are listed in table D.12 (discard all patterns with $a_3 \neq 0$). The pattern values are graphed in diagram D.7. Intersecting the bold lines, corresponding to patterns 2 and 9 gives $\frac{1}{2}V + 1 = \frac{5}{2}V + \frac{1}{2}$, so that the minimum value of $\mathcal{R}(V)$ is attained for $V = \frac{1}{4}$, which yields a ratio of $\frac{9}{8}$. Substitution of $V = \frac{1}{4}$ in (D18) gives the weights as used in (8.45).

	a_1	a_2	a_3	a_4	value	case 6a V= 1/4	case 6b V= 1/4
1	2	0	0	0	1	24/24	8/8
2	1	1	0	1	1/2V+1	27/24	→ 9/8 ←
3	1	0	2	0	7/6	→ 28/24 ←	—
4	1	0	1	1	V+5/6	26/24	—
5	1	0	0	2	2V+1/2	24/24	8/8
6	0	2	0	1	1	24/24	8/8
7	0	1	2	0	-1/2V+7/6	25/24	—
8	0	1	1	1	1/2V+5/6	23/24	—
9	0	1	0	3	5/2V+1/2	27/24	→ 9/8 ←
10	0	0	3	0	1	24/24	—
11	0	0	2	2	2V+2/3	→ 28/24 ←	—
12	0	0	1	3	3V+1/3	26/24	—
13	0	0	0	4	4V	24/24	8/8

Table D.12. Extremal patterns of program (D17)
Generator, case 6a: $\mathcal{A} = \{10a_1 + 9a_2 + 7a_3 + 5a_4 \leqslant 24\}$
Generator, case 6b: $\mathcal{A} = \{\ 6a_1 + 5a_2 \qquad + 3a_4 \leqslant 14\}$

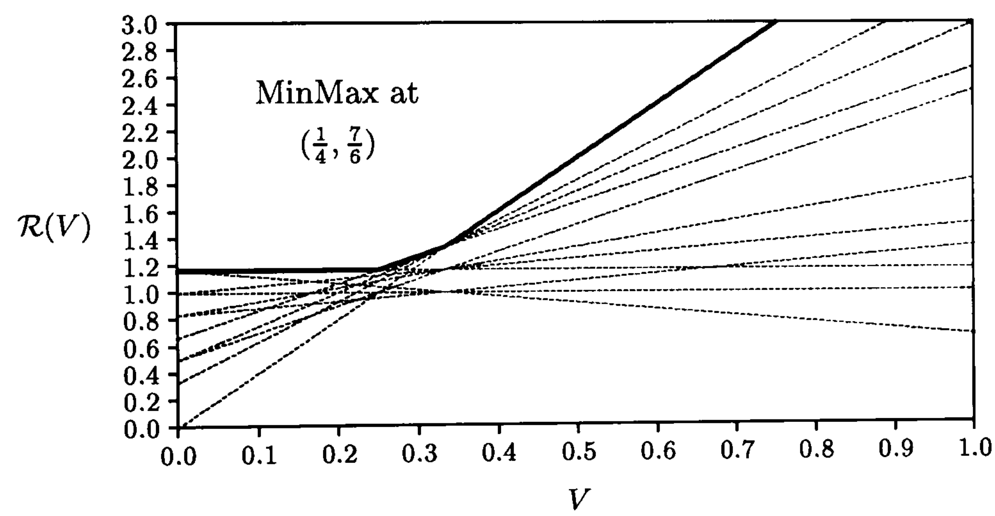


Diagram D.6. Pattern values for program (D17).

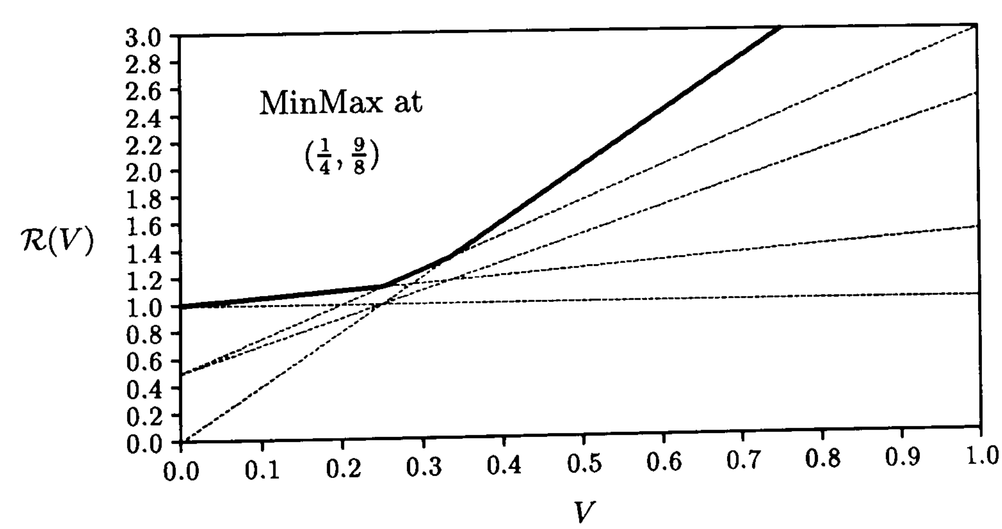


Diagram D.7. Pattern values for program (D19).

D.1.7 Further analysis of $x \in \langle \frac{1}{4}, \frac{1}{5} \rangle$

In this subsection we will perform a more detailed analysis on two of the configurations in section 8.8. The aim is to sharpen some of the bounds derived there and thus eliminate some possible cases of (FFD, OPT).

D.1.7.1 $v \in \langle \frac{1}{2} - x, \frac{1}{3}]$

We will prove that the following bound holds for any list which packs in a configuration as shown in diagram 8.19. This is a tighter bound then the one implied by (8.35) and allows the elimination of the case $(FFD, OPT) = (13, 10)$.

$$x \in \langle \tfrac{1}{5}, \tfrac{1}{5}] \text{ and } v \in \langle \tfrac{1}{2} - x, \tfrac{1}{3}] \;\Rightarrow\; FFD \leqslant \tfrac{7}{8} + \tfrac{29}{24} OPT \qquad \text{(D20)}$$

- First note that when we are considering a bound in OPT that the weight of item w in (8.34) can be increased to 5/8. This follows along similar lines as the original derivation of the weight of item w (section D.1.2, p. 187), the only difference being that, although the pattern (w, w) is valid for CSP_R, it does not correspond to a valid bin in a packing by OPT since there is only one item w. This means that we can increase $W(w)$ to $29/24 - W(y) - W(x) = 5/8$. This gives the bound $FFD \leqslant 11/12 + W(\mathcal{L}) = 11/12 + \sum_{j=1}^{OPT} W_j$.
- We may assume that $\delta_2 = 1$ (diagram 8.19), otherwise the bound $FFD \leqslant 3/4 + (29/24)OPT$ holds.
- If there is a bin in the optimal packing with a weight strictly less than 29/24 then the bound $FFD < 11/12 + (29/24)OPT$ holds, which implies (D20).

For the remainder we assume that all bins have a weight of exactly 29/24 (obviously, no bin can have a weight exceeding 29/24) and show that this leads to a contradiction. Table D.3 shows that there are only two ways of combining items of weights 15/24, 9/24, 8/24 and 6/24 to a [bin]weight of 29/24. This gives exactly seven possibilities for the bin configurations that the items y_1 and y_2 can pack into.

$$[v, y, y_1, y_2], [v, y, y_1, x], [v, y, y_2, x] \qquad \text{(D21)}$$

$$[z, y, y_1], [z, y, y_2] \text{ and } [w, y, y_1], [w, y, y_2] \qquad \text{(D22)}$$

To eliminate the patterns in (D21) it suffices to show that $[v, y, y_2, x]$ is infeasible. This pattern has a length that exceeds 1 because $y + y_1 + y_2 > 1 - x$ holds by virtue of the FFD-rule and $v \geqslant y_1$. The other two patterns have a larger length, and are thus also infeasible. This means that y_1 and y_2 must pack into two different bins, which must be among the configurations listed in (D22). The total length of the items packed in these two bins is at least $z + w + 2y + y_1 + y_2 > 1 + 2y + y_1 + y_2 > 1 + y + y_1 + y_2 + x > 2$, so that there is no combination of patterns in (D22) that is feasible.

This contradicts the assumption that all bins have a weight of exactly 29/24 and thus proves the bound (D20).

D.1.7.2 $v \in \langle \frac{1-x}{3}, \frac{1}{2} - x]$

We will prove that the following bound holds for any list which packs into a configuration as shown in diagram 8.20. That there are instances that achieve this bound follows from diagrams D.8 and D.9.

$$x \in \langle \tfrac{1}{5}, \tfrac{1}{4}] \text{ and } v \in \langle \tfrac{1-x}{3}, \tfrac{1}{2} - x] \quad \Rightarrow \quad FFD \leqslant \tfrac{62}{81} + \tfrac{11}{9} CSP_R \tag{D23}$$

The first observation we can make is that we may assume $\delta_1 = 1$, otherwise the bound $FFD \leqslant \frac{3}{4} + \frac{11}{9} CSP_R$ would hold. At this point it is not obvious how one should condition on the sizes in order to use the weighting-function approach to eventually arrive at bound (D23). Instead we use a more direct approach. We know that the minimal configuration originates from a list $\mathcal{L}$ that has the following structure

$$\mathcal{L} = \{ \underbrace{z, \ldots, z}_{n_1}, \underbrace{v, \ldots, v}_{n_1}, \underbrace{y, \ldots, y}_{3n_2+1}, y_1, y_2, \underbrace{x, \ldots, x}_{4n_3+1} \}, \tag{D24}$$

and is packed in $FFD(\mathcal{L}) = n_1 + n_2 + n_3 + 2$ bins. The sizes of the items, $\mathbf{s}^{\mathsf{T}} = [z, v, y, y_1, y_2, x]$ satisfy the following

$$\begin{array}{cccc} z \in \langle \frac{1}{2}, 1 - 2x] & v \in \langle \frac{1-x}{3}, \frac{1}{2} - x] & y \in \langle \frac{1-x}{3}, v] & x \in \langle \frac{1}{5}, \frac{1}{4} \rangle \\ & z + v > 1 - x & y + y_1 + y_2 > 1 - x & \\ & & z \geqslant v \geqslant y \geqslant y_1 \geqslant y_2 \geqslant x & \end{array} \tag{D25}$$

We want to find the maximum value of $c(\mathcal{L})$ given by the expression $c(\mathcal{L}) = FFD(\mathcal{L}) - (11/9) CSP_R(\mathcal{L})$. To this end we formulate the following mixed integer program

$$c(\mathbf{s}) = \boxed{\begin{array}{ll} \text{Max} & n_1 + n_2 + n_3 + 2 - \frac{11}{9} \sum x_j \\ \text{st} & \sum \mathbf{a}_j x_j \geqslant \mathbf{f} \\ & n_i \in \mathbb{N},\ \mathbf{a}_j \in \mathcal{A}(\mathbf{s}) \text{ and } x_j \geqslant 0 \end{array}} \tag{D26}$$

where

$$\mathbf{f}^{\mathsf{T}} = [n_1, n_1, 3n_2 + 1, 1, 1, 4n_3 + 1] \tag{D27}$$

and

$$\mathcal{A}(\mathbf{s}) = \left\{ \mathbf{a} \in \mathbb{N}^6 \mid z a_1 + v a_2 + y a_3 + y_1 a_4 + y_2 a_5 + x a_6 \leqslant 1 \right\}. \tag{D28}$$

This program gives the maximum value of $c(\mathcal{L})$ for a given set of item sizes. To see this, note that for any fixed $\mathbf{n}$ the problem reduces to a cutting stock problem over the items in the list $\mathcal{L}$ and thus $\sum x_j = CSP_R(\mathcal{L})$. Since $FFD(\mathcal{L}) = n_1 + n_2 + n_3 + 2$ we have that $c(\mathbf{s})$ gives the maximum value of $c(\mathcal{L})$ for a given $\mathbf{s}$. It also shows that (D26) has a finite solution-value by (8.37); viz. $c(\mathbf{s}) \leqslant 11/12$.

The problem now lies in determining the item sizes that maximise $c(\mathbf{s})$, or the equivalent, to determine the pattern set $\mathcal{A}(\mathbf{s})$ for which $c(\mathbf{s})$ is maximised.

Containing set for $\mathcal{A}(\mathbf{s})$ Rather than determining the pattern set $\mathcal{A}(\mathbf{s})$ for a specific instance of the item sizes we determine a larger set $\mathcal{A}$. This set is determined such that $\mathcal{A}(\mathbf{s}) \subset \mathcal{A}$ for all possible instances s. This set is defined as follows

$$\forall \mathbf{s} \quad \mathcal{A}(\mathbf{s}) \subset \mathcal{A} \equiv \left\{ \mathbf{a} \in \mathbb{N}^6 \mid za_1 + va_2 + ya_3 + xa_4 + xa_5 + xa_6 \leqslant 1 \right\}. \tag{D29}$$

In determining the ratio for configuration 8.20 we have already enumerated the extremal-pattern set for the items z, v, y and x (see table D.5, p. 190). A generator for this set, and thus for $\mathcal{A}$, was determined as $(z, v, y, x) = \frac{1}{44}(23, 13, 12, 9)$. This gives the following explicit definition. We note that $\mathcal{A}$ has 67 extremal patterns.

$$\mathcal{A} = \left\{ \mathbf{a} \in \mathbb{N}^6 \mid 23a_1 + 13a_2 + 12a_3 + 9a_4 + 9a_5 + 9a_6 \leqslant 44 \right\}. \tag{D30}$$

Pruning of $\mathcal{A}$ Using the conditions on the item sizes, listed in (D25), means that we can eliminate some patterns from $\mathcal{A}$, since they cannot be in any of the sets $\mathcal{A}(\mathbf{s})$. For instance

$$\begin{aligned}
y + y_1 + y_2 > 1 - x &\quad \Rightarrow \quad \text{delete pattern } [0,0,1,1,1,1] \\
v + y_1 + y_2 > 1 - x &\quad \Rightarrow \quad \text{delete pattern } [0,1,0,1,1,1] \\
2y + y_1 > 1 - x &\quad \Rightarrow \quad \text{delete pattern } [0,0,2,1,0,1] \\
2v + y_1 > 1 - x &\quad \Rightarrow \quad \text{delete pattern } [0,2,0,1,0,1] \\
&\quad \textit{etc.}
\end{aligned}$$

After checking all the [extremal] patterns in the set $\mathcal{A}$ for feasibility, that is does there exist an s such that this pattern is in $\mathcal{A}(\mathbf{s})$, and eliminating all those that cannot be in any $\mathcal{A}(\mathbf{s})$ we end up with the set $\mathcal{A}_0 = \cup_{\mathbf{s}} \mathcal{A}(\mathbf{s})$. This set has 45 extremal patterns, which are listed in table D.13.

We now return to solving (D26), and relax the constraint $\mathbf{a}_j \in \mathcal{A}(\mathbf{s})$ to $\mathbf{a}_j \in \mathcal{A}_0$. Solving the relaxation gives a solution as shown in (D31) with solution value 7/9.

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 66 \\ 22 \\ 33 \end{bmatrix}, \quad \mathbf{x}_B = \begin{bmatrix} 66 \\ 32\frac{5}{6} \\ \frac{1}{3} \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix} \quad \text{and} \quad \begin{matrix} z \\ v \\ y \\ y_1 \\ y_2 \\ x \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 3 & 0 & 0 \\ 1 & 2 & 0 & 2 & 1 \end{bmatrix} \mathbf{x}_B = \mathbf{f} = \begin{bmatrix} 66 \\ 66 \\ 67 \\ 1 \\ 1 \\ 133 \end{bmatrix} \tag{D31}$$

$$\left. \begin{aligned} FFD = n_1 + n_2 + n_3 + 2 &= 122 \\ CSP_R = \sum x_j &= 100 \end{aligned} \right\} \Rightarrow c \leqslant 122 - \tfrac{11}{9} \times 100 = \tfrac{7}{9} \tag{D32}$$

Conditioning on item sizes The solution in (D31) now gives a clue on how to condition on the item sizes. Although each of the patterns $\mathbf{a}_j$ in the optimal LP-solution is individually feasible, that is there exists an s such that $\mathbf{a}_j \in \mathcal{A}(\mathbf{s})$, it need not be that a combination of patterns is feasible. This is the case for the following two patterns in the optimal basis-matrix in (D31).

$$\left.\begin{array}{l}\text{pattern 5: } y_1 \leqslant \frac{1-x}{3} \\ \text{pattern 3: } y_2 \leqslant \frac{1-v}{3}\end{array}\right\} \Rightarrow y + y_1 + y_2 \leqslant \frac{1-x}{3} + \frac{1-v}{3} + y \leqslant \frac{1-x}{3} + \frac{1+2v}{3} \leqslant 1 - x \qquad \text{(D33)}$$

Patterns 3 and 5 cannot both be valid since this implies that $y + y_1 + y_2 \leqslant 1 - x$ and FFD would have placed another item in the transition bin (diagram 8.20). This means that pattern $[0, 0, 1, 1, 1, 0]$ cannot represent a feasible transition-bin. We therefore need to condition on the sizes, such as to eliminate one of the offending patterns, 3 or 5. The most obvious way to do this is to condition on the size of item y_1. Note however that we have already established the bound $FFD \leqslant \frac{7}{9} + \frac{11}{9} CSP_R$.

$y_1 > \frac{1-x}{3}$ This condition knocks out pattern $[0, 0, 0, 3, 0, 1]$. Deleting this pattern from $\mathcal{A}_0$ gives $\mathcal{A}_1$ with 45 extremal patterns (see table D.13). Solving the MIP on $\mathcal{A}_1$ gives a maximum value for c of $62/81 \approx 0.76543...$ for the patterns as shown in (D34).

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 71 \\ 23 \\ 35 \end{bmatrix}, \ \mathbf{x}_B = \begin{bmatrix} 70 \\ 1 \\ \frac{2}{9} \\ 35 \\ \frac{1}{3} \end{bmatrix} \quad \text{and} \quad \begin{matrix} z \\ v \\ y \\ y_1 \\ y_2 \\ x \end{matrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 2 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 1 & 1 & 0 & 2 & 0 \end{bmatrix} \mathbf{x}_B = \mathbf{f} = \begin{bmatrix} 71 \\ 71 \\ 70 \\ 1 \\ 1 \\ 141 \end{bmatrix} \qquad \text{(D34)}$$

$$\left.\begin{array}{rl} FFD = n_1 + n_2 + n_3 + 2 & = 131 \\ CSP_R = \sum x_j & = 106\frac{5}{9} \end{array}\right\} \Rightarrow c \leqslant 131 - \tfrac{11}{9} \times 106\tfrac{5}{9} = \tfrac{62}{81} \qquad \text{(D35)}$$

We now need to show that this combination of patterns corresponds to a feasible instance of s. Since all these patterns (and the FFD-patterns) are extremal patterns we use the procedure described in appendix B.7 to find a generator. This gives an instance with sizes $\mathbf{s}^{\mathsf{T}} = \frac{1}{94}[48, 28, 27, 27, 22, 19]$. The list and corresponding packings are given in diagram D.8.

$y_1 \leqslant \frac{1-x}{3}$ As a consequence of this condition we have $y_2 > \frac{1-y}{3}$, since

$$y_2 \leqslant \tfrac{1-y}{3} \Rightarrow y + y_1 + y_2 \leqslant y + \tfrac{1-x}{3} + \tfrac{1-y}{3} = \tfrac{1-x}{3} + \tfrac{1+2y}{3} \leqslant 1 - x \qquad \text{(D36)}$$

This knocks out pattern $[0, 0, 1, 0, 3, 0]$. Deleting this pattern from $\mathcal{A}_0$ gives $\mathcal{A}_2$ with 43 extremal patterns (see table D.13). Solving the MIP on $\mathcal{A}_2$ gives a maximum value for c of $62/81 \approx 0.76543...$ for the patterns as shown in (D37).

$$\begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} = \begin{bmatrix} 71 \\ 23 \\ 35 \end{bmatrix}, \ \mathbf{x}_{\mathrm{B}} = \begin{bmatrix} 70 \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{2}{9} \\ 35\frac{1}{6} \\ \frac{1}{6} \end{bmatrix} \quad \text{and} \quad \begin{matrix} z \\ v \\ y \\ y_1 \\ y_2 \\ x \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 1 \end{bmatrix} \mathbf{x}_{\mathrm{B}} = \mathbf{f} = \begin{bmatrix} 71 \\ 71 \\ 70 \\ 1 \\ 1 \\ 141 \end{bmatrix} \tag{D37}$$

$$\left.\begin{aligned} FFD = n_1 + n_2 + n_3 + 2 &= 131 \\ CSP_R = \sum x_j &= 106\tfrac{5}{9} \end{aligned}\right\} \Rightarrow \ c \leqslant 131 - \tfrac{11}{9} \times 106\tfrac{5}{9} = \tfrac{62}{81} \tag{D38}$$

To show that this combination of patterns corresponds to a feasible instance of **s**, we again use the procedure described in appendix B.7 to find a generator. This gives an instance with sizes $\mathbf{s}^{\mathsf{T}} = \frac{1}{154}[78, 46, 45, 41, 38, 31]$. The list and corresponding packings are given in diagram D.9;

Conclusion The case analysis for y_1 proves that in both cases the bound $c(\mathrm{s}) \leqslant 62/81$ holds and thus proves the bound (D23).

Instance for $x \in \langle \frac{2}{11}, \frac{1}{5}]$ We note that by perturbing the instance in diagram D.8 we have constructed a list with $FFD = 22/27 + (11/9)\,CSP_R$ (see diagram D.10).

List $\mathcal{L} = \{ \underbrace{48, \ldots, 48}_{6k+5}, \underbrace{28, \ldots, 28}_{6k+5}, \underbrace{27, \ldots, 27}_{6k+5}, 22, \underbrace{19, \ldots, 19}_{12k+9} \}$ on bins of size 94.

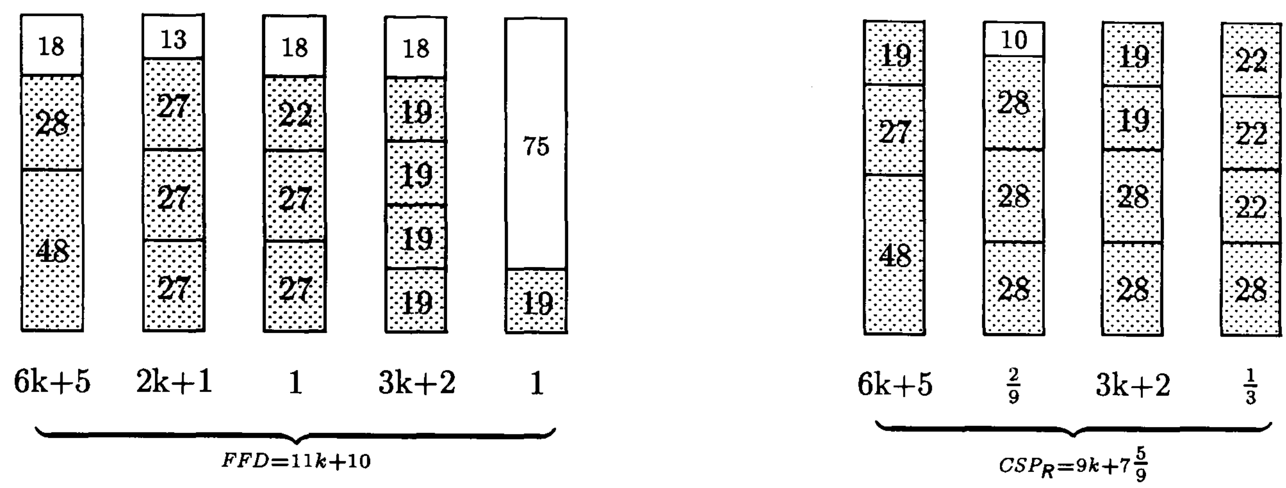


Diagram D.8. Example for $FFD = \frac{62}{81} + \frac{11}{9} CSP_R$.

List $\mathcal{L} = \{ \underbrace{78, \ldots, 78}_{6k+5}, \underbrace{46, \ldots, 46}_{6k+5}, \underbrace{45, \ldots, 45}_{6k+4}, 41, 38, \underbrace{31, \ldots, 31}_{12k+9} \}$ on bins of size 154.

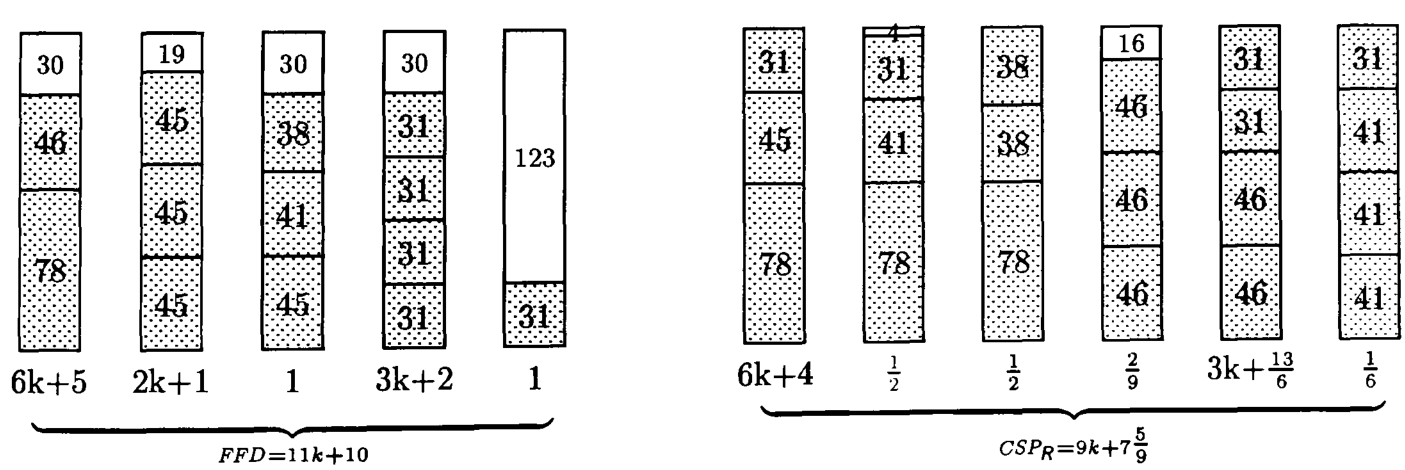


Diagram D.9. Example for $FFD = \frac{62}{81} + \frac{11}{9} CSP_R$

	a_1	a_2	a_3	a_4	a_5	a_6		a_1	a_2	a_3	a_4	a_5	a_6
1	1	1	0	0	0	0	23	0	0	3	0	0	0
2	1	0	1	0	0	1	24	0	0	2	1	0	0
3	1	0	0	1	0	1	25	0	0	2	0	1	0
4	1	0	0	0	2	0	26	0	0	2	0	0	2
5	1	0	0	0	1	1	27	0	0	1	2	0	0
6	1	0	0	0	0	2	28	0	0	1	1	1	0
7	0	3	0	0	0	0	29	0	0	1	1	0	2
8	0	2	1	0	0	0	30	0	0	1	0	3	0
9	0	2	0	1	0	0	31	0	0	1	0	2	1
10	0	2	0	0	1	0	32	0	0	1	0	1	2
11	0	2	0	0	0	2	33	0	0	1	0	0	3
12	0	1	2	0	0	0	34	0	0	0	3	0	1
13	0	1	1	1	0	0	35	0	0	0	2	1	1
14	0	1	1	0	1	0	36	0	0	0	2	0	2
15	0	1	1	0	0	2	37	0	0	0	1	3	0
16	0	1	0	2	0	0	38	0	0	0	1	2	1
17	0	1	0	1	1	0	39	0	0	0	1	1	2
18	0	1	0	1	0	2	40	0	0	0	1	0	3
19	0	1	0	0	3	0	41	0	0	0	0	4	0
20	0	1	0	0	2	1	42	0	0	0	0	3	1
21	0	1	0	0	1	2	43	0	0	0	0	2	2
22	0	1	0	0	0	3	44	0	0	0	0	1	3
							45	0	0	0	0	0	4

Table D.13. Extremal pattern sets for the $FFD = \frac{62}{81} + \frac{11}{9} CSP_R$ example.

- patterns listed are the extremal patterns of $\mathcal{A}_0 = \cup_{\mathbf{s}} \mathcal{A}(\mathbf{s})$.
- To get the extremal pattern set for case 1, replace pattern 34 by $[0,0,0,3,0,0]$. Generated by $\mathcal{A}_1 = \{\mathbf{a} \in \mathbb{N}^6 \mid 63a_1 + 37a_2 + 36a_3 + 35a_4 + 29a_5 + 25a_6 \leqslant 124\}$.
- To get the extremal pattern set for case 2, delete patterns 19 and 30. Generated by $\mathcal{A}_2 = \{\mathbf{a} \in \mathbb{N}^6 \mid 93a_1 + 55a_2 + 54a_3 + 49a_4 + 45a_5 + 37a_6 \leqslant 184\}$.

List $\mathcal{L} = \{ \underbrace{48, \ldots, 48}_{6k+5}, \underbrace{28, \ldots, 28}_{6k+5}, \underbrace{27, \ldots, 27}_{6k+5}, 22, \underbrace{19, \ldots, 19}_{12k+10}, 18.8, 18.8, 18.5 \}$ on bins of size 94.

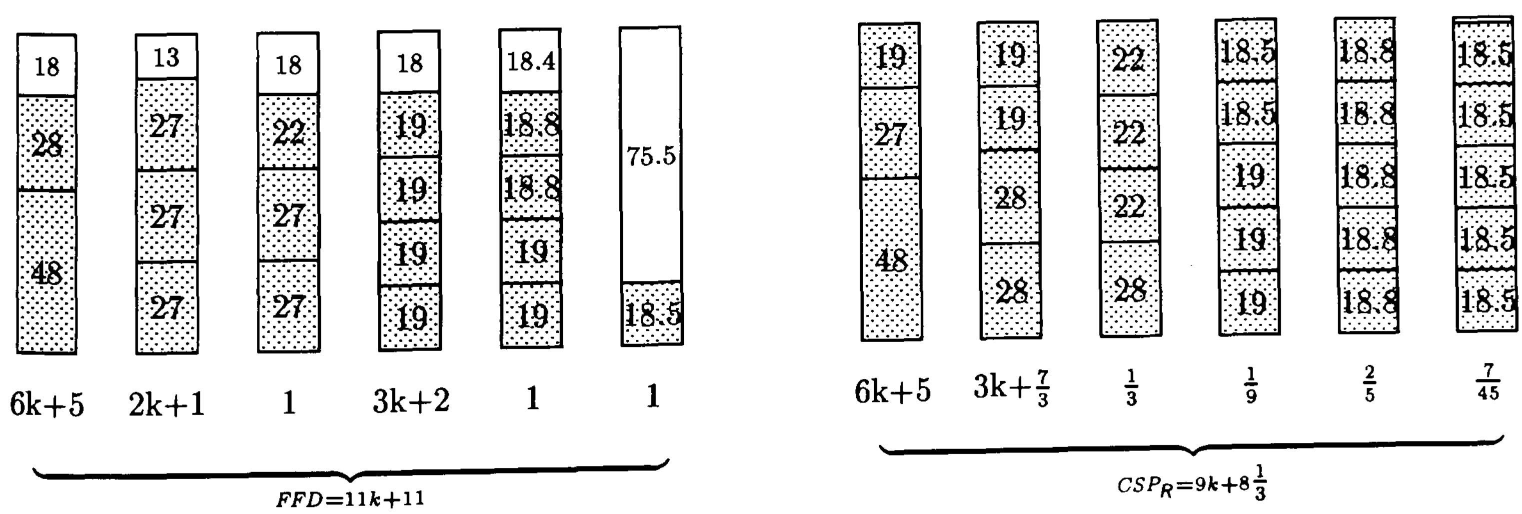


Diagram D.10. Example for $FFD = \frac{22}{27} + \frac{11}{9} CSP_R$.

D.2 $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right]$

D.2.1 $x \in \left\langle \frac{1}{6}, \frac{1}{5} \right]$ and no 1-item

We start with the generic weighting function for $x \in \left\langle \frac{1}{6}, \frac{1}{5} \right\rangle$, with $V \in \left[\frac{1}{5}, \frac{1}{4}\right]$ to be determined. This range of V ensures that $W(s)$ is non decreasing.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{1-V}{2}, & s \in \left\langle \frac{1}{3}, \frac{1-x}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right] \\ \frac{1-V}{3}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ V, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \xrightarrow[V = 1/5]{\text{Minimising value}} W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{2}{5}, & s \in \left\langle \frac{1}{3}, \frac{1-x}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right] \\ \frac{4}{15}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{1}{5}, & s \in \left[x, \frac{1-x}{4} \right] \end{cases} \tag{D39}$$

Consider the decision problem.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + \frac{1-V}{3}a_4 + \frac{1}{4}a_5 + Va_6 \\ \text{st} & \left(\frac{1-x}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + \left(\frac{1}{4}\right)^+ a_4 + \left(\frac{1-x}{4}\right)^+ a_5 + xa_6 \leqslant 1 \\ & x \in \left\langle \frac{1}{6}, \frac{1}{5} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D40}$$

From appendix C we know that $\mathcal{R}(V) = \max\{\mathcal{R}_0(V), \mathcal{R}_1(V)\}$, where $\mathcal{R}_0(V)$ and $\mathcal{R}_1(V)$ are the knapsack problems for the boundary values of x.

$$\mathcal{R}_0(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + \frac{1-V}{3}a_4 + \frac{1}{4}a_5 + Va_6 \\ \text{st} & 30a_1 + 24a_2 + 20a_3 + 18a_4 + 15a_5 + 12a_6 \leqslant 71 \quad (x = \frac{1}{6}) \\ & a_i \in \mathbb{N} \end{array}} \tag{D41}$$

$$\mathcal{R}_1(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + \frac{1-V}{3}a_4 + \frac{1}{4}a_5 + Va_6 \\ \text{st} & 24a_1 + 20a_2 + 16a_3 + 15a_4 + 12a_5 + 12a_6 \leqslant 59 \quad (x = \frac{1}{5}) \\ & a_i \in \mathbb{N} \end{array}} \tag{D42}$$

We solve each of the knapsack problems, following the procedure outlined in appendix C. The value of $\mathcal{R}(V)$ for the boundary values $x = \frac{1}{6}$ and $x = \frac{1}{5}$ is given in tables D.14 and D.15. These tables are to be read as follows. The first column gives the range for which the pattern in the following columns is an optimal solution to $\mathcal{R}(V)$. The last column gives the solution value of $\mathcal{R}(V)$ for the specified range. Graphing both tables we get diagram D.11. Intersecting the lines $\frac{7}{12} + 3V$ and $\frac{5}{4} - \frac{1}{3}V$ gives a minimising value $V = \frac{1}{5}$ with corresponding value of $\mathcal{R}(\frac{1}{5}) = \frac{71}{60}$. There are exactly three patterns, listed in table D.16, that achieve the value of $\frac{71}{60}$. Substitution of $V = \frac{1}{5}$ now gives the weighting function as used in section 8.9.2.

V-range	a_1	a_2	a_3	a_4	a_5	a_6	$\mathcal{R}_0(V)$	$V^* = \frac{1}{5}$
$[0, \frac{1}{6}]$	0	2	1	0	0	0	$\frac{4}{3} - V$	$\frac{68}{60}$
$[\frac{1}{6}, \frac{1}{5}]$	0	1	1	0	1	1	$\frac{13}{12} + \frac{1}{2}V$	$\rightarrow \frac{71}{60} \leftarrow$
$[\frac{1}{5}, \frac{1}{4}]$	0	0	1	0	1	3	$\frac{7}{12} + 3V$	$\rightarrow \frac{71}{60} \leftarrow$
$[\frac{1}{4}, \frac{1}{3}]$	0	0	1	0	0	4	$\frac{1}{3} + 4V$	$\frac{68}{60}$
$[\frac{1}{3}, 1]$	0	0	0	0	0	5	$5V$	$\frac{60}{60}$

Table D.14. Solution of ratio problem (D41)

V-range	a_1	a_2	a_3	a_4	a_5	a_6	$\mathcal{R}_1(V)$	$V^* = \frac{1}{5}$
$[0, \frac{1}{6}]$	1	1	0	1	0	0	$\frac{4}{3} - \frac{5}{6}V$	$\frac{70}{60}$
$[\frac{1}{6}, \frac{1}{4}]$	0	0	2	1	1	0	$\frac{5}{4} - \frac{1}{3}V$	$\rightarrow \frac{71}{60} \leftarrow$
$[\frac{1}{4}, \frac{1}{3}]$	0	0	2	0	0	2	$\frac{2}{3} + 2V$	$\frac{64}{60}$
$[\frac{1}{3}, 1]$	0	0	0	0	0	4	$4V$	$\frac{48}{60}$

Table D.15. Solution of ratio problem (D42)

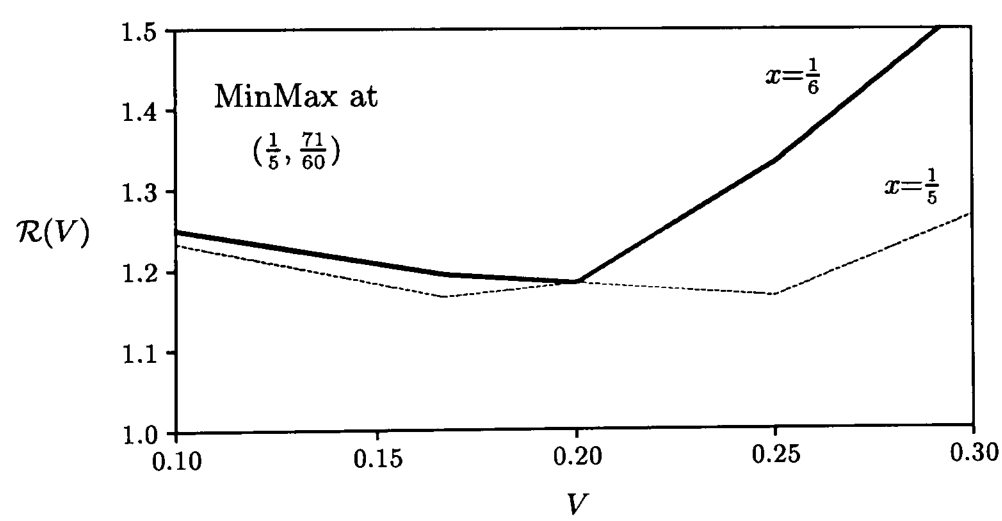


Diagram D.11. Solution value of ratio problem (D40)

	a_1	a_2	a_3	a_4	a_5	a_6	value	valid for
1	0	1	1	0	1	1	$\frac{13}{12} + \frac{1}{2}V$	$x < \frac{1}{5}$
2	0	0	2	1	1	0	$\frac{5}{4} - \frac{1}{3}V$	$x > \frac{2}{11}$
3	0	0	1	0	1	3	$\frac{7}{12} + 3V$	$x < \frac{5}{29}$

Table D.16. Optimal patterns of program (D40) for $V = \frac{1}{5}$

Transition patterns

In the previous sections we could enumerate the transition patterns more or less by hand because we were dealing with relatively small problems. However we now need a more structured approach.
As usual we characterise a pattern by the number of items it has in the intervals that we partioned into. The set $\mathcal{A}(x)$ represents all feasible patterns for a given value of x.

$$\mathcal{A}(x) \equiv \left\{ \mathbf{a} \in \mathbb{N}^6 \mid \left(\tfrac{1-x}{2}\right)^+ a_1 + \left(\tfrac{1}{3}\right)^+ a_2 + \left(\tfrac{1-x}{3}\right)^+ a_3 + \left(\tfrac{1}{4}\right)^+ a_4 + \left(\tfrac{1-x}{4}\right)^+ a_5 + x a_6 \leqslant 1 \right\} \tag{D43}$$

Rather than determining this set for each value of x, we determine the set $\mathcal{A} = \bigcup_x \mathcal{A}(x)$. This set can be expressed as $\mathcal{A} = \mathcal{A}(\frac{1}{6}) \cup \mathcal{A}(\frac{1}{5})$ by virtue of lemma C3.
All bins (except the [last] singleton-bin) are filled to a level of at least $1 - x$. A superset for the patterns that represent a FFD-bin (excluding the singleton bin) for a given value of x is defined by $\mathcal{A}_H(x)$.

$$\mathcal{A}_H(x) \equiv \left\{ \mathbf{a} \in \mathcal{A} \mid \tfrac{1}{2} a_1 + \tfrac{1-x}{2} a_2 + \tfrac{1}{3} a_3 + \tfrac{1-x}{3} a_4 + \tfrac{1}{4} a_5 + \tfrac{1-x}{4} a_6 > 1 - x \right\} \tag{D44}$$

Rather than determining this set for each value of x, we determine $\mathcal{A}_H = \bigcup_x \mathcal{A}_H(x) = \mathcal{A}_H(\frac{1}{6}) \cup \mathcal{A}_H(\frac{1}{5})$.

We now want to determine the set of all possible configurations for a (transition) bin with bin weight strictly less than one. This set is contained in the set $\mathcal{A}_T$ defined as follows.

$$\mathcal{A}_T \equiv \{ \mathbf{a} \in \mathcal{A} \mid \mathbf{a} \in \mathcal{A}_H \text{ and } W(\mathbf{a}) < 1 \} \tag{D45}$$

The set $\mathcal{A}_T$ can now be determined by the following algorithm, given in pseudo code.

```
A_T := ∅; x_0 := 1/6; x_1 := 1/5
FOR x ∈ {x_0, x_1} DO { enumerate A(x) }
    FOR a ∈ A(x) DO
      IF (a ∈ A_H(x_0) or a ∈ A_H(x_1)) and (W(a) < 1)
      THEN A_T := a ∪ A_T
```

This gives a set of 28 possible configurations for the transition bins. These are listed in table D.17. Note that we have excluded the singleton bin from the table.

Set-packing problem

Solving the corresponding set-packing problem gives a value of $\frac{29}{60}$ for pattern combination $(2, 18, 28)$. This yields an upper bound for the constant of $c \leqslant \frac{29}{60} + (1 - \frac{1}{5}) = \frac{77}{60}$.

The weighting functions for subsequent cases are stronger weighting functions than the one used for this case. This implies that the set of patterns defining the corresponding SPP is a subset of the patterns in table D.17.

Balance

Using optimal pattern 3 (table D.16) we set up a balance in table D.18. The symbols u, y and x denote the sizes as in diagram 8.37.

Generator

To find actual sizes we could enumerate the extremal patterns of program (D40). Instead we use the patterns in table D.18 to give the requirements for the sizes and set up an IP as follows.

$$\begin{array}{llcl} \text{Min} & L & & \\ \text{st} & 3u \leqslant L & \text{and} & 3u + x \geqslant L + 1 \\ & 4y \leqslant L & \text{and} & 4y + x \geqslant L + 1 \\ & 5x \leqslant L & \text{and} & 6x \geqslant L + 1 \\ & & u + y + 3x \leqslant L & \\ & L, u, y, x \in \mathbb{N} & & \end{array} \tag{D46}$$

This gives as a solution $(L, u, y, x) = (71, 20, 15, 12)$. Note that these values correspond to the constraint of program (D41). For this program to be feasible $x/L < 5/29$ must hold.

Extension of weighting function

For $x \in \langle \frac{2}{11}, \frac{1}{5}]$ we extend the generic weighting function (D39) (with $V = \frac{1}{5}$) to deal with the case when there are 1-items in the list. Consider the following programs;

$$\begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i < 1 - 3x \\ & x_i \geqslant 0 \end{array} \quad \text{and} \quad \begin{array}{ll} \text{Max} & \sum W(x_i) \\ \text{st} & \sum x_i < \frac{1}{2} \\ & x_i \geqslant 0 \end{array} \tag{D47}$$

which have a value of $\frac{1}{2}$ and $\frac{7}{12}$ respectively. This means that, without increasing the ratio of $\frac{71}{60}$, we can extend the weighting function (D39) (with $V = 1/5$) as follows.

$$W(s) = \begin{cases} \frac{2}{3}, & s \in \langle 3x, 1 - 2x] \\ \frac{3}{5}, & s \in \langle \frac{1}{2}, 3x] \end{cases} \tag{D48}$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		*																*										*
$W_T \times 60$	54	50	58	57	54	59	56	56	55	52	54	51	56	55	52	52	51	50	59	56	59	56	58	55	52	57	54	51
$(1 - W_T) \times 60$	6	10	2	3	6	1	4	4	5	8	6	9	4	5	8	8	9	10	1	4	1	4	2	5	8	3	6	9
$\langle\frac{1-x}{2}, \frac{1}{2}]$ 30/60	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle\frac{1}{3}, \frac{1-x}{2}]$ 24/60	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle\frac{1-x}{3}, \frac{1}{3}]$ 20/60	**0**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle\frac{1}{4}, \frac{1-x}{3}]$ 16/60	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**2**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**2**	**1**	**0**	**0**	**0**	**2**	**2**	**1**	**1**	**1**	**0**	**0**	**0**
$\langle\frac{1-x}{4}, \frac{1}{4}]$ 15/60	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**2**	**1**	**0**	**1**	**0**	**0**	**1**	**2**	**1**	**0**	**1**	**0**	**2**	**1**	**0**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$ 12/60	**0**	**0**	**1**	**1**	**2**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**2**	**3**	**1**	**2**	**1**	**2**	**3**	**1**	**2**	**3**
	1	1	1	1	1																							
		1	1	1	1	1	1	1	1	1	1	1																
			1	1	1								1	1	1	1	1	1	1	1								
				1	1																1	1	1	1	1			
				1	1																					1	1	1
								1	1	1	1	1	1	1	1	1	1	1	1	1								
									1	1	1	1									1	1	1	1	1			
										1		1														1	1	1
														1	1		1	1	1	1	1	1	1	1	1			
																						1		1	1	1	1	1

Table D.17. Transition patterns and SPP-formulation for configuration 8.37, $x \in \langle 1/6, 1/5]$. Shown are all patterns with a length $> 1 - x$ and a bin weight strictly less than one under the generic weighting function (8.18). Pattern combination $\{2, 18, 28\}$ represents the optimum solution to the corresponding set-packing problem.

	FFD-bins					LP-patterns
u	**3**	**0**	**0**	**0**	60k+12	**1**
y	**0**	**4**	**0**	**0**	60k+12	**1**
x	**0**	**0**	**5**	**1**	180k+36	**3**
	20k+4	15k+3	36k+7	1		60k+12
	71k+15					**60k+12**

Table D.18. Balance for $FFD = \frac{4}{5} + \frac{71}{60} OPT$

D.2.2 $x \in \langle\frac{2}{11},\frac{1}{5}\rangle$ and there is a 1-item $z > 3x$

We only need to consider the cases with $v \leqslant \frac{1-x}{3}$. The analysis of these cases starts with the following weights and weighting function.

$$W(z)=V,\ W(v)=1-V \text{ and } W(s)=\begin{cases}\frac{3}{5}, & s\in\langle\frac{1}{2},3x]\\ \frac{1}{2}, & s\in\langle\frac{1-x}{2},\frac{1}{2}]\\ \frac{2}{5}, & s\in\langle 2x,\frac{1-x}{2}]\\ \frac{4}{15}, & s\in\langle\frac{1}{4},\frac{1-x}{3}]\\ \frac{1}{4}, & s\in\langle\frac{1-x}{4},\frac{1}{4}]\\ \frac{1}{5}, & s\in[x,\frac{1-x}{4}]\end{cases} \tag{D49}$$

Note that since there are no items in $\langle v, 2x]$ there are certainly no items in $\langle\frac{1-x}{3},2x]$. Note further that this weighting function is recurrent.

D.2.2.1 Ratio problem for $v \leqslant \frac{1-x}{3}$

An upper bound for the ratio, using this weighting function, is given by the following program.

$$\mathcal{R}(V)=\begin{array}{|ll|}\hline \text{Max} & Va_1+(1-V)a_2+\frac{3}{5}a_3+\frac{1}{2}a_4+\frac{2}{5}a_5+\frac{4}{15}a_6+\frac{1}{4}a_7+\frac{1}{5}a_8\\ \text{st} & za_1+va_2+\left(\frac{1}{2}\right)^+a_3+w_1a_4+w_2a_5+\left(\frac{1}{4}\right)^+a_6+\left(\frac{1-x}{4}\right)^+a_7+xa_8\leqslant 1\\ & z+v>1-x\\ & z+w_1>1,\quad \text{for } w_1\in\langle\frac{1-x}{2},\frac{1}{2}]\\ & z+w_2>1,\quad \text{for } w_2\in\langle 2x,\frac{1-x}{2}]\\ & \text{further constraints}\\ & x\in\langle\frac{2}{11},\frac{1}{5}\rangle,\ v\in\langle\text{case-range}],\ z\in\langle 3x,1-2x] \text{ and } a_i\in\mathbb{N}\\ \hline\end{array} \tag{D50}$$

The 'further constraints' depend upon the particular range of v that we condition on. Using the fact that the list contains no items in $\langle v, 2x]$ we can simplify the program by adding the following constraints.

$$\begin{array}{ll} \text{case } 3,4: & v\in\langle\frac{1-x}{4},\frac{1}{4}] \Rightarrow a_6=0\\ \text{case } 5\quad : & v\in\langle x,\frac{1-x}{4}] \Rightarrow a_6=a_7=0\end{array}$$

We will now replace constraints 2–4, which are in terms of the item sizes, by constraints in terms of the decision variables a_1, a_2, a_4 and a_5.

- Without altering the value of $\mathcal{R}(V)$ we can add the constraint $5a_1 + a_2 \leqslant 5$. This has the effect of forcing $a_2 = 0$ whenever $a_1 = 1$. The solution $a_1 = a_2 = 1$ implies $a_3 = \cdots = a_8 = 0$, since $z + v > 1 - x$ and has solution value 1. Since there is another solution with value 1, viz. pattern $5e_9$, we can exclude the former solution without affecting the value of $\mathcal{R}(V)$. The constraint $5a_1 + a_2 \leqslant 5$ excludes pattern $\mathbf{a} = e_1 + e_2$ from the solution set. It does not exclude any other pattern since $a_2 \leqslant \lfloor 1/v\rfloor \leqslant \lfloor 1/x\rfloor \leqslant \lfloor 11/2\rfloor = 5$, by the first constraint in program (D50).

- The constraint $z + w_1 > 1$ implies the cut $2a_1 + a_4 \leqslant 2$. It is easily verified that this cut does not exclude any feasible solution, since $a_1 \leqslant 1$, $a_4 \leqslant 2$ and $a_1 = 1$ implies $a_4 = 0$.
- The constraint $z + w_2 > 1$ implies the cut $2a_1 + a_5 \leqslant 2$. It is easily verified that this cut does not exclude any feasible solution, since $a_1 \leqslant 1$, $a_5 \leqslant 2$ and $a_1 = 1$ implies $a_5 = 0$.

We can now add the cuts, delete the corresponding constraints and replace $w_{1,2}$ by their lower bounds. The upper bound for z is not essential for the ratio problem, but the lower bounds are. These can be combined into one requirement. This finally gives the following relaxation.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & za_1 + va_2 + \left(\frac{1}{2}\right)^+ a_3 + \left(\frac{1-x}{2}\right)^+ a_4 + (2x)^+ a_5 + \left(\frac{1}{4}\right)^+ a_6 + \left(\frac{1-x}{4}\right)^+ a_7 + xa_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & \text{further constraints} \\ & x \in \left\langle \frac{2}{11}, \frac{1}{5} \right\rangle,\ v \in \langle \text{case-range}],\ z > \max\{3x, 1-x-v\} \text{ and } a_i \in \mathbb{N} \end{array}} \qquad \text{(D51)}$$

We note that the route taken to relax the ratio problem (D50) $\rightarrow$ (D51) does not depend upon the actual weighting function used. Therefore choosing a different [read stronger] weighting function leads to the same [relaxation of the] ratio problem.

Case specific ratio problem To derive the specific ratio problem for each individual case we substitute values for the item sizes z and v, depending upon the case in question.

$$\begin{array}{llll} \text{case 2}: & v \in \langle \frac{1}{4}, \frac{1-x}{3}] & \Rightarrow\ z > 3x & \text{and } v > \frac{1}{4} \\ \text{case 3}: & v \in \langle \frac{1-3x}{2}, \frac{1}{4}] & \Rightarrow\ z > 3x & \text{and } v > \frac{1-3x}{2} \\ \text{case 4}: & v \in \langle \frac{1-x}{4}, \frac{1-3x}{2}] & \Rightarrow\ z > \frac{1+x}{2} & \text{and } v > \frac{1-x}{4} \\ \text{case 5}: & v \in \langle x, \frac{1-x}{4}] & \Rightarrow\ z > \frac{3}{4}(1-x) & \text{and } v > x \end{array}$$

Now substitute the corresponding lower bound for z and v in the first constraint, delete all other requirements on z and v and we have arrived at a program with constraints solely depending upon x.

D.2.2.2 Constant

The weights used for the items in block 3 (which determine the constant) are exactly the same as (or larger than) the weights we have used for the case when there are no 1-items (section 8.9.2). This means that the constant c has the same upper bound; that is $c \leqslant \frac{77}{60}$. However, to derive a tighter upper bound we solve a separate set-packing problem for each individual case. The formulation of these SPPs is facilitated by the fact that the defining patterns are a subset of the patterns in table D.17.

D.2.2.3 $x \in \langle\frac{2}{11}, \frac{1}{5}\rangle$, $z > 3x$ and $v \in \langle\frac{1}{4}, \frac{1-x}{3}]$

Ratio Problem Substituting the lower bounds for z and v in program (D51), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & (3x)^+ a_1 + \left(\frac{1}{4}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + \left(\frac{1-x}{2}\right)^+ a_4 + (2x)^+ a_5 \\ & \qquad + \left(\frac{1}{4}\right)^+ a_6 + \left(\frac{1-x}{4}\right)^+ a_7 + xa_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & x \in \langle\frac{2}{11}, \frac{1}{5}\rangle \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D52}$$

If $a_1, \ldots, a_7 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$3xa_1 + \tfrac{1}{4}a_2 + \tfrac{1}{2}a_3 + \tfrac{1-x}{2}a_4 + 2xa_5 + \tfrac{1}{4}a_6 + \tfrac{1-x}{4}a_7 + xa_8 < 1 \tag{D53}$$

Substituting the extremal values of x, gives the following constraints.

$$\begin{cases} 24a_1 + 11a_2 + 22a_3 + 18a_4 + 16a_5 + 11a_6 + 9a_7 + 8a_8 \leqslant 43, & (x = \frac{2}{11}) \\ 36a_1 + 15a_2 + 30a_3 + 24a_4 + 24a_5 + 15a_6 + 12a_7 + 12a_8 \leqslant 59, & (x = \frac{1}{5}) \end{cases} \tag{D54}$$

The constraint for $x = \frac{1}{5}$ is implied by the constraint for $x = \frac{2}{11}$, so that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & 24a_1 + 11a_2 + 22a_3 + 18a_4 + 16a_5 + 11a_4 + 9a_5 + 8a_6 \leqslant 43 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & a_i \in \mathbb{N} \end{array}} \tag{D55}$$

Solving this program for different values of V gives table D.19. This directly gives $\frac{11}{16}$ as the minimising value for V, which gives a value $\mathcal{R}(\frac{11}{16}) = \frac{19}{16}$. There are two (optimal) patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of the items corresponding to $\left(\frac{1}{2}\right)^+$, $(2x)^+$, $\left(\frac{1}{4}\right)^+$ and x without increasing the ratio. These can be increased to $\frac{20}{32}$, $\frac{14}{32}$, $\frac{9}{32}$ and $\frac{7}{32}$ respectively. This finally gives the weighting function (8.54).

Constant The set-packing problem, under this (stronger) weighting function, is given in table D.20. It has a value of $\frac{7}{32}$ for pattern combination $(1, 6, 12)$. An upper bound for the constant therefore is $c \leqslant \frac{7}{32} + 1 - \frac{7}{32} = 1$.

Problem Instance A balance with sizes is given in table D.21.

V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\mathcal{R}(V)$	$V^* = \frac{11}{16}$	valid for	
$[0, \frac{11}{16}]$	**0**	**3**	**0**	**0**	**0**	**0**	**1**	**0**	$\frac{13}{4} - 3V$	$\rightarrow \frac{19}{16} \leftarrow$	$x > 0$	$v < \frac{3+x}{12}$
$[\frac{11}{16}, 1]$	**1**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	$\frac{1}{2} + V$	$\rightarrow \frac{19}{16} \leftarrow$	$x < \frac{1}{5}$	$v > \frac{1-3x}{2}$

Table D.19. Solution of ratio problem (D55)

		*					*						*
		1	2	3	4	5	6	7	8	9	10	11	12
$W_T \times 32$		30	31	30	31	30	30	29	31	30	31	30	29
$(1 - W_T) \times 32$		2	1	2	1	2	2	3	1	2	1	2	3
$\langle \frac{1-x}{2}, \frac{1}{2}]$	16/32	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle 2x, \frac{1-x}{2}]$	14/32	**1**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1}{4}, \frac{1-x}{3}]$	9/32	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1-x}{4}, \frac{1}{4}]$	8/32	**0**	**1**	**0**	**1**	**0**	**2**	**1**	**1**	**0**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	7/32	**0**	**1**	**2**	**0**	**1**	**0**	**1**	**2**	**3**	**1**	**2**	**3**
		1	1	1									
			1	1	1	1	1	1					
			1	1					1	1			
			1	1							1	1	1
					1	1	1	1	1	1			
						1		1			1	1	1
									1	1	1	1	1

Table D.20. SPP-formulation under weighting function (8.54)

	FFD-bins				LP-patterns		
$z = \frac{22}{38}$	**1**	**0**	**0**	9+12k	**1**	**0**	**0**
$v = \frac{10}{38}$	**1**	**0**	**0**	9+12k	**0**	**3**	**3**
$y = \frac{8}{38}$	**0**	**4**	**0**	20+28k	**2**	**1**	**0**
$x = \frac{7}{38}$	**0**	**0**	**1**	1	**0**	**0**	**1**
	9+12k	5+7k	1		9+12k	2+4k	1
	15+19k				**12+16k**		

Table D.21. Balance for $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$.

D.2.2.4 $x \in \langle \frac{2}{11}, \frac{1}{5} \rangle$, $z > 3x$ and $v \in \langle \frac{1-3x}{2}, \frac{1}{4}]$

Ratio Problem Substituting the lower bounds for z and v in program (D51), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \begin{array}{|ll|} \hline \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{4}a_6 + \frac{1}{5}a_7 \\ \text{st} & (3x)^+ a_1 + (\frac{1-3x}{2})^+ a_2 + (\frac{1}{2})^+ a_3 + (\frac{1-x}{2})^+ a_4 + (2x)^+ a_5 + (\frac{1-x}{4})^+ a_6 + xa_7 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & x \in \langle \frac{2}{11}, \frac{1}{5} \rangle \text{ and } a_i \in \mathbb{N} \\ \hline \end{array} \qquad \text{(D56)}$$

If $a_1, \ldots, a_6 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$3xa_1 + \tfrac{1-3x}{2}a_2 + \tfrac{1}{2}a_3 + \tfrac{1-x}{2}a_4 + 2xa_5 + \tfrac{1-x}{4}a_6 + xa_7 < 1 \qquad \text{(D57)}$$

Substituting the extremal values of x, gives the following constraints.

$$\begin{cases} 24a_1 + 10a_2 + 22a_3 + 18a_4 + 16a_5 + 9a_6 + 8a_7 \leqslant 43, & (x = \frac{2}{11}) \\ 6a_1 + 2a_2 + 5a_3 + 4a_4 + 4a_5 + 2a_6 + 2a_7 \leqslant 9, & (x = \frac{1}{5}) \end{cases} \qquad \text{(D58)}$$

The constraint for $x = \frac{1}{5}$ is implied by the constraint for $x = \frac{2}{11}$, so that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \begin{array}{|ll|} \hline \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{4}a_6 + \frac{1}{5}a_7 \\ \text{st} & 24a_1 + 10a_2 + 22a_3 + 18a_4 + 16a_5 + 9a_6 + 8a_7 \leqslant 43 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & a_i \in \mathbb{N} \\ \hline \end{array} \qquad \text{(D59)}$$

Solving this program for different values of V gives table D.22. This table directly gives $\frac{7}{10}$ as the minimising value for V, which gives a value $\mathcal{R}(\frac{7}{10}) = \frac{6}{5}$. There are three (optimal) patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of the items corresponding to $(\frac{1-x}{2})^+$, $(2x)^+$ and x without increasing the ratio. These can be increased to $\frac{31}{60}$, $\frac{26}{60}$ and $\frac{13}{60}$ respectively. This gives the weighting function (8.56)

Constant The set-packing problem, under this (stronger) weighting function, is given in table D.23. It has a value of $\frac{13}{60}$ for pattern combination $(1,4,8)$. An upper bound for the constant therefore is $c \leqslant \frac{13}{60} + 1 - \frac{13}{60} = 1$.

Problem Instance A balance with sizes is given in table D.24.

V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{7}{10}$	valid for	
$[0, \frac{7}{10}]$	**0**	**4**	**0**	**0**	**0**	**0**	**0**	$4 - 4V$	$\rightarrow \frac{6}{5} \leftarrow$	$x > \frac{1}{6}$	$v \leqslant \frac{1}{4}$
$[\frac{7}{10}, \frac{7}{10}]$	**0**	**2**	**1**	**0**	**0**	**0**	**0**	$\frac{13}{5} - 2V$	$\rightarrow \frac{6}{5} \leftarrow$	$x > \frac{1}{6}$	$v < \frac{1}{4}$
$[\frac{7}{10}, 1]$	**1**	**0**	**0**	**0**	**0**	**2**	**0**	$\frac{1}{2} + V$	$\rightarrow \frac{6}{5} \leftarrow$	$x < \frac{1}{5}$	$v > \frac{1-3x}{2}$

Table D.22. Solution of ratio problem (D59)

		*			*				*
		1	2	3	4	5	6	7	8
$W_T \times 60$		57	59	57	56	54	58	56	54
$(1 - W_T) \times 60$		3	1	3	4	6	2	4	6
$\langle \frac{1-x}{2}, \frac{1}{2}]$	31/60	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
$\langle 2x, \frac{1-x}{2}]$	26/60	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1-x}{4}, \frac{1}{4}]$	15/60	**0**	**1**	**0**	**2**	**1**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	13/60	**0**	**1**	**2**	**0**	**1**	**1**	**2**	**3**
		1	1	1					
			1	1	1	1			
			1	1			1	1	1
						1	1	1	1

Table D.23. SPP-formulation under weighting function (8.56)

	FFD-bins				LP-patterns			
$z = \frac{28}{48}$	**1**	**0**	**0**	1+4k	**1**	**0**	**1**	**0**
$v = \frac{12}{48}$	**1**	**0**	**0**	1+4k	**0**	**4**	**0**	**1**
$y = \frac{10}{48}$	**0**	**4**	**0**	4+8k	**2**	**0**	**1**	**3**
$x = \frac{9}{48}$	**0**	**0**	**1**	1	**0**	**0**	**1**	**0**
	1+4k	1+2k	1		4k	k	1	1
	3+6k				**2+5k**			

Table D.24. Balance for $FFD = \frac{3}{5} + \frac{6}{5} CSP_R$.

D.2.2.5 $x \in \langle \frac{2}{11}, \frac{1}{5} \rangle$, $z > 3x$ and $v \in \langle \frac{1-x}{4}, \frac{1-3x}{2}]$

Ratio Problem Substituting the lower bounds for z and v in program (D51), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \begin{array}{|ll|}\hline \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{4}a_6 + \frac{1}{5}a_7 \\ \text{st} & \left(\frac{1+x}{2}\right)^+ a_1 + \left(\frac{1-x}{4}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + \left(\frac{1-x}{2}\right)^+ a_4 + (2x)^+ a_5 + \left(\frac{1-x}{4}\right)^+ a_6 + xa_7 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & x \in \langle \frac{2}{11}, \frac{1}{5} \rangle \text{ and } a_i \in \mathbb{N} \\ \hline \end{array} \tag{D60}$$

If $a_1, \ldots, a_6 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by;

$$\tfrac{1+x}{2}a_1 + \tfrac{1-x}{4}a_2 + \tfrac{1}{2}a_3 + \tfrac{1-x}{2}a_4 + 2xa_5 + \tfrac{1-x}{4}a_6 + xa_7 < 1 \tag{D61}$$

Substituting the extremal values of x, gives the following constraints;

$$\begin{cases} 26a_1 + 9a_2 + 22a_3 + 18a_4 + 16a_5 + 9a_6 + 8a_7 \leqslant 43, & (x = \frac{2}{11}) \\ 6a_1 + 2a_2 + 5a_3 + 4a_4 + 4a_5 + 2a_6 + 2a_7 \leqslant 9, & (x = \frac{1}{5}) \end{cases} \tag{D62}$$

The constraint for $x = \frac{1}{5}$ is implied by the constraint for $x = \frac{2}{11}$. This is easily verified by scaling both to have the same RHS. So that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \begin{array}{|ll|}\hline \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{4}a_6 + \frac{1}{5}a_7 \\ \text{st} & 26a_1 + 9a_2 + 22a_3 + 18a_4 + 16a_5 + 9a_5 + 8a_6 \leqslant 43 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & a_i \in \mathbb{N} \\ \hline \end{array} \tag{D63}$$

Solving this program for different values of V gives table D.25. This table directly gives $\frac{59}{80}$ as the minimising value for V, which gives a value $\mathcal{R}(\frac{59}{80}) = \frac{19}{16}$. There are three (optimal) patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of the items corresponding to $\left(\frac{1}{2}\right)^+$ and $\left(\frac{1-x}{2}\right)^+$ without increasing the ratio. These can be increased to $\frac{53}{80}$ and $\frac{42}{80}$ respectively. This gives the weighting function (8.58).

Constant The set-packing problem under this (stronger) weighting function is given in table D.26. It has a value of $\frac{26}{80}$ for pattern combination (1, 4, 8). This gives an upper bound for the constant of $c \leqslant \frac{26}{80} + 1 - \frac{16}{80} = \frac{9}{8}$.

Problem Instance A balance with sizes is given in table D.27.

V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{59}{80}$	valid for	
$[0, \frac{3}{5}]$	**0**	**4**	**0**	**0**	**0**	**0**	**0**	$4 - 4V$	$\frac{84}{80}$		
$[\frac{3}{5}, \frac{59}{80}]$	**0**	**3**	**0**	**0**	**1**	**0**	**0**	$\frac{17}{5} - 3V$	$\rightarrow \frac{95}{80} \leftarrow$	$x < \frac{1}{5}$	$v < \frac{1-2x}{3}$
	0	**3**	**0**	**0**	**0**	**0**	**2**	$\frac{17}{5} - 3V$	$\rightarrow \frac{95}{80} \leftarrow$	$x < \frac{1}{5}$	$v \leqslant \frac{1-2x}{3}$
$[\frac{59}{80}, 1]$	**1**	**0**	**0**	**0**	**0**	**1**	**1**	$\frac{9}{20} + V$	$\rightarrow \frac{95}{80} \leftarrow$	$x < \frac{1}{5}$	$v > \frac{1-x}{4}$

Table D.25. Solution of ratio problem (D63)

		*			*				*
		1	2	3	4	5	6	7	8
$W_T \times 80$		74	78	74	72	68	76	72	68
$(1 - W_T) \times 80$		6	2	6	8	12	4	8	12
$\langle \frac{1-x}{2}, \frac{1}{2}]$	42/80	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
$\langle 2x, \frac{1-x}{2}]$	32/80	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1-x}{4}, \frac{1}{4}]$	20/80	0	**1**	0	**2**	**1**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	16/80	0	**1**	**2**	0	**1**	**1**	**2**	**3**
		1	1	1					
			1	1	1	1			
			1	1			1	1	1
						1	1	1	1

Table D.26. SPP-formulation under weighting function (8.58)

	FFD-bins					LP-patterns		
$z = \frac{104}{170}$	**1**	**0**	**0**	**0**	9+12k	**1**	**0**	**1**
$v = \frac{36}{170}$	**1**	**0**	**0**	**0**	9+12k	**0**	**3**	**0**
$y = \frac{35}{170}$	**0**	**4**	**0**	**0**	8+12k	**1**	**0**	**0**
$x = \frac{31}{170}$	**0**	**0**	**5**	**1**	16+20k	**1**	**2**	**2**
	9+12k	2+3k	3+4k	1		8+12k	3+4k	1
	15+19k					**12+16k**		

Table D.27. Balance for $FFD = \frac{3}{4} + \frac{19}{16} CSP_R$.

D.2.2.6 $x \in \langle\frac{2}{11}, \frac{1}{5}\rangle$, $z > 3x$ and $v \in \langle x, \frac{1-x}{4}]$

Ratio Problem Substituting the lower bounds for z and v in program (D51), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{5}a_6 \\ \text{st} & \frac{3}{4}(1-x)^+ a_1 + x^+ a_2 + (\frac{1}{2})^+ a_3 + (\frac{1-x}{2})^+ a_4 + (2x)^+ a_5 + xa_6 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & x \in \langle\frac{2}{11}, \frac{1}{5}\rangle \text{ and } a_i \in \mathbb{N} \end{array}} \qquad \text{(D64)}$$

If $a_1, \ldots, a_5 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by;

$$\tfrac{3}{4}(1-x)a_1 \;+\; xa_2 \;+\; \tfrac{1}{2}a_3 \;+\; \tfrac{1-x}{2}a_4 \;+\; 2xa_5 \;+\; xa_6 \;< 1 \qquad \text{(D65)}$$

Substituting the extremal values of x, gives the following constraints;

$$\begin{cases} 27a_1 + 8a_2 + 22a_3 + 18a_4 + 16a_5 + 8a_6 \leqslant 43, & (x = \frac{2}{11}) \\ 6a_1 + 2a_2 + 5a_3 + 4a_4 + 4a_5 + 2a_6 \leqslant 9, & (x = \frac{1}{5}) \end{cases} \qquad \text{(D66)}$$

The constraint for $x = \frac{1}{5}$ is implied by the constraint for $x = \frac{2}{11}$. This is easily verified by scaling both to have the same RHS. So that $\mathcal{R}(V)$ simplifies to

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{2}{5}a_5 + \frac{1}{5}a_6 \\ \text{st} & 27a_1 + 8a_2 + 22a_3 + 18a_4 + 16a_5 + 8a_6 \leqslant 43 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \text{ and } 2a_1 + a_5 \leqslant 2 \\ & a_i \in \mathbb{N} \end{array}} \qquad \text{(D67)}$$

Solving this program for different values of V gives table D.28. This table directly gives $\frac{31}{40}$ as the minimising value for V, which gives a value $\mathcal{R}(\frac{31}{40}) = \frac{47}{40}$. There are two (optimal) patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of the items corresponding to $(\frac{1}{2})^+$ and $(2x)^+$ without increasing the ratio. These can be increased to $\frac{27}{40}$ and $\frac{18}{40}$ respectively. This gives the weighting function (8.60)

Constant The set-packing problem, under this (stronger) weighting function, is given in table D.29. It has a value of $\frac{4}{40}$ for pattern 2. An upper bound for the constant therefore is $c \leqslant \frac{4}{40} + 1 - \frac{8}{40} = \frac{9}{10}$.

Problem Instance A balance with sizes is given in table D.30.

V-range	a_1	a_2	a_3	a_4	a_5	a_6	$\mathcal{R}(V)$	$V^* = \frac{31}{40}$	valid for	
$[0, \frac{3}{4}]$	**0**	**5**	**0**	**0**	**0**	**0**	$5 - 5V$	$\frac{45}{40}$		
$[\frac{3}{4}, \frac{31}{40}]$	**0**	**3**	**0**	**1**	**0**	**0**	$\frac{7}{2} - 3V$	$\rightarrow \frac{47}{40} \leftarrow$	$x < \frac{1}{5}$	$v < \frac{1+x}{6}$
$[\frac{31}{40}, 1]$	**1**	**0**	**0**	**0**	**0**	**2**	$\frac{2}{5} + V$	$\rightarrow \frac{47}{40} \leftarrow$	$x < \frac{1}{5}$	$v > x$

Table D.28. Solution of ratio problem (D67)

			*
		1	2
$W_T \times 40$		38	36
$(1 - W_T) \times 40$		2	4
$\langle \frac{1-x}{2}, \frac{1}{2}]$	20/40	**1**	**1**
$\langle 2x, \frac{1-x}{2}]$	18/40	**1**	**0**
$[x, \frac{1-x}{4}]$	8/40	**0**	**2**
		1	1

Table D.29. SPP-formulation under weighting function (8.60)

	FFD-bins					LP-patterns	
$z = \frac{52}{82}$	**1**	0	0	0	18+30k	**1**	0
$w = \frac{34}{82}$	0	**2**	0	0	6+10k	0	**1**
$v = \frac{16}{82}$	**1**	0	0	0	18+30k	0	**3**
$x = \frac{15}{82}$	0	0	**5**	**1**	36+60k	**2**	0
	18+30k	3+5k	7+12k	1		18+30k	6+10k
	29+47k					**24+40k**	

Table D.30. Balance for $FFD = \frac{4}{5} + \frac{47}{40} CSP_R$.

D.2.3 $x \in \langle \frac{2}{11}, \frac{1}{5}]$ and all 1-items $\leqslant 3x$

We only need to consider the cases with $v \leqslant \frac{1}{3}$. The analysis of these cases starts with the following weights and weighting function.

$$W(z) = V,\ W(v) = 1 - V \text{ and } W(s) = \begin{cases} \frac{3}{5}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{1}{2}, & s \in \langle 1-3x, \frac{1}{2}] \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1}{3}] \\ \frac{4}{15}, & s \in \langle \frac{1}{4}, \frac{1-x}{3}] \\ \frac{1}{4}, & s \in \langle \frac{1-x}{4}, \frac{1}{4}] \\ \frac{1}{5}, & s \in [x, \frac{1-x}{4}] \end{cases} \tag{D68}$$

Note that since there are no items in $\langle v, 1-3x]$ there are certainly no items in $\langle \frac{1}{3}, 1-3x]$. Note further that this weighting function is recurrent.

D.2.3.1 Ratio problem (for $v \leqslant \frac{1}{3}$)

An upper bound for the ratio, using this weighting function, is given by the following program.

$$\mathcal{R}(V) = \begin{array}{|ll|} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & za_1 + va_2 + (\frac{1}{2})^+ a_3 + wa_4 + (\frac{1-x}{3})^+ a_5 + (\frac{1}{4})^+ a_6 + (\frac{1-x}{4})^+ a_7 + xa_8 \leqslant 1 \\ & z + v > 1 - x \\ & z + w > 1 \quad \text{for } w \in \langle 1-3x, \frac{1}{2}] \\ & \text{further constraints} \\ & x \in \langle \frac{2}{11}, \frac{1}{5}],\ v \in \langle\text{case-range}],\ z \in \langle \frac{1}{2}, 3x] \text{ and } a_i \in \mathbb{N} \end{array} \tag{D69}$$

The 'further constraints' depend upon the particular range of v that we condition on. Using the fact that the list contains no items in $\langle v, 1-3x]$ we can simplify the program by adding the following constraints.

$$\text{case 5}: \ v \in \langle 1-4x, \tfrac{1-x}{3}] \ \Rightarrow \ a_5 = 0$$

By the same rationale as in section D.2.2.1 we can replace the constraints $z + v > 1 - x$ and $z + w > 1$ by the constraints $5a_1 + a_2 \leqslant 5$ and $2a_1 + a_4 \leqslant 2$, respectively.
This gives the following relaxation.

$$\mathcal{R}(V) = \begin{array}{|ll|} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & za_1 + va_2 + (\frac{1}{2})^+ a_3 + (1-3x)^+ a_4 + (\frac{1-x}{3})^+ a_5 + (\frac{1}{4})^+ a_6 + (\frac{1-x}{4})^+ a_7 + xa_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \\ & \text{further constraints} \\ & x \in \langle \frac{2}{11}, \frac{1}{5}],\ v \in \langle\text{case-range}],\ z > \max\{\frac{1}{2}, 1-x-v\} \text{ and } a_i \in \mathbb{N} \end{array} \tag{D70}$$

We note that the route taken to relax the ratio problem (D69) $\rightarrow$ (D70) does *not* depend upon the actual weighting function used. Therefore choosing a different [read stronger] weighting function leads to the same [relaxation of the] ratio problem.

Case specific ratio problem To derive the specific ratio problem for each individual case we substitute values for the item sizes z and v, depending upon the case in question.

$$\begin{array}{llll}
\text{case } 2: & v \in \langle \frac{1+x}{4}, \frac{1}{3}] & \Rightarrow\ z > \frac{1}{2} & \text{and } v > \frac{1+x}{4} \\
\text{case } 3: & v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}] & \Rightarrow\ z > \frac{3-5x}{4} & \text{and } v > \frac{7-19x}{12} \\
\text{case } 4: & v \in \langle \frac{1-x}{3}, \frac{7-19x}{12}] & \Rightarrow\ z > \frac{5+7x}{12} & \text{and } v > \frac{1-x}{3} \\
\text{case } 5: & v \in \langle 1-4x, \frac{1-x}{3}] & \Rightarrow\ z > \frac{2}{3}(1-x) & \text{and } v > 1-4x
\end{array}$$

Now substitute the corresponding lower bound for z and v in the first constraint, delete all other requirements on z and v and we have arrived at a program with constraints solely depending upon x.

D.2.3.2 Constant

For block 3 and 4 we have the same weighting function as for the case when there is no 1-item, section 8.9.2. This means that we can use the same set-packing problem to derive an upper bound for the constant and thus that $c \leqslant \frac{77}{60}$.

D.2.3.3 $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z \leqslant 3x$ and $v \in \langle \frac{1+x}{4}, \frac{1}{3}]$

We substitute the lower bounds for z and v in program (D70), and delete all other constraints in z and v. Furthermore, it turns out that the constraints corresponding to $z+v > 1-x$ and $z+w > 1$ are not necessary. This gives the following program.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & \left(\frac{1}{2}\right)^+ a_1 + \left(\frac{1+x}{4}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 \\ & \qquad + \left(\frac{1-x}{3}\right)^+ a_5 + \left(\frac{1}{4}\right)^+ a_6 + \left(\frac{1-x}{4}\right)^+ a_7 + xa_8 \leqslant 1 \\ & x \in \langle \frac{2}{11}, \frac{1}{5}] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D71}$$

If $a_1, \ldots, a_7 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$\tfrac{1}{2}a_1 \ + \ \tfrac{1+x}{4}a_2 \ + \ \tfrac{1}{2}a_3 \ + \ \big(1-3x\big)a_4 \ + \ \tfrac{1-x}{3}a_5 \ + \ \tfrac{1}{4}a_6 \ + \ \tfrac{1-x}{4}a_7 \ + \ xa_8 \ < 1 \tag{D72}$$

Substituting the extremal values of x, gives the following constraints.

$$\begin{cases} 22a_1 + 13a_2 + 22a_3 + 20a_4 + 12a_5 + 11a_6 + \ \ 9a_7 + \ \ 8a_8 \leqslant 43, & (x = \frac{2}{11}) \\ 30a_1 + 18a_2 + 30a_3 + 24a_4 + 16a_5 + 15a_6 + 12a_7 + 12a_8 \leqslant 59, & (x = \frac{1}{5}) \end{cases} \tag{D73}$$

Solving the resulting IP for the extremal values of x gives table D.31. From this table we can see that the minimising value for V is $\frac{5}{8}$, which gives a value $\mathcal{R}(\frac{5}{8}) = \frac{29}{24}$. There are three patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of items corresponding to $\left(\frac{1}{2}\right)^+$, $\left(\frac{1}{4}\right)^+$ and x without increasing the ratio. These can be increased to $\frac{5}{8}$, $\frac{7}{24}$ and $\frac{5}{24}$ respectively. This gives the weighting function (8.64).

Constant The set-packing problem, under this (stronger) weighting function, is given in table D.32. It has a value of $\frac{9}{24}$ for pattern combinations $(1, 9, 19)$ and $(1, 13, 19)$. An upper bound for the constant therefore is $c \leqslant \frac{9}{24} + 1 - \frac{5}{24} = \frac{7}{6}$.

Problem Instance A balance with sizes is given in table D.33.

Sharpening of constant It turns out that the bound resulting from the above constant is too weak; we cannot prove bound (8.1) with it. In order to sharpen the bound we analyse the SPP more closely, in particular the first transition-bin.

- If pattern 1 (table D.32) is not active then SPP has a value of $\frac{6}{24}$ for pattern combinations $(9, 19)$ and $(13, 19)$. This gives an upper bound for the constant of $c \leqslant \frac{6}{24} + 1 - \frac{5}{24} = \frac{25}{24}$.
- If there is a bin with an item $> \frac{1}{2}$ and an item $> 1 - 3x$, it has a weight of $\frac{5}{8} + \frac{1}{2} = \frac{9}{8}$. This gives an upper bound for the constant of $\frac{7}{6} + (1 - \frac{9}{8}) = \frac{25}{24}$.

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\mathcal{R}(V)$	$V^* = \frac{5}{8}$	valid for	
$x = \frac{2}{11}$	$[0, \frac{11}{20}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{135}{120}$		
	$[\frac{11}{20}, \frac{37}{60}]$	**0**	**2**	**0**	**0**	**0**	**0**	**1**	**1**	$\frac{49}{20} - 2V$	$\frac{144}{120}$		
	$[\frac{37}{60}, \frac{5}{8}]$	**0**	**1**	**0**	**0**	**1**	**0**	**2**	**0**	$\frac{11}{6} - V$	$\rightarrow \frac{145}{120} \leftarrow$	$x > \frac{1}{7}$	$v < \frac{1+5x}{6}$
	$[\frac{5}{8}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	$\frac{7}{12} + V$	$\rightarrow \frac{145}{120} \leftarrow$		
$x = \frac{1}{5}$	$[0, \frac{7}{12}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{135}{120}$		
	$[\frac{7}{12}, \frac{5}{8}]$	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	$\frac{11}{6} - V$	$\rightarrow \frac{145}{120} \leftarrow$	$x > \frac{7}{37}$	$v < \frac{10x-1}{3}$
	$[\frac{5}{8}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	$\frac{7}{12} + V$	$\rightarrow \frac{145}{120} \leftarrow$	$x > \frac{1}{7}$	$v > \frac{7-19x}{12}$

Table D.31. Solution of ratio problem (D71)

		*												*						*
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$W_T \times 24$		21	23	22	23	23	22	23	22	21	23	22	22	21	23	23	22	23	22	21
$(1 - W_T) \times 24$		3	1	2	1	1	2	1	2	3	1	2	2	3	1	1	2	1	2	3
$\langle 1 - 3x, \frac{1}{2}]$	12/24	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1+x}{4}, \frac{1}{3}]$	9/24	**1**	**0**	**0**	**2**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1-x}{3}, \frac{1+x}{4}]$	8/24	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**2**	**2**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1}{4}, \frac{1-x}{3}]$	7/24	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**1**	**0**	**1**	**0**	**2**	**1**	**0**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1-x}{4}, \frac{1}{4}]$	6/24	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**2**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	5/24	**0**	**1**	**2**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**2**	**3**	**1**	**2**	**3**
		1	1	1																
					1	1	1	1	1	1	1	1	1	1	1					
																1	1	1	1	1
			1	1														1	1	1
			1	1												1	1			

Table D.32. SPP-formulation under weighting function (8.64)

	FFD-bins					LP-patterns		
$z = \frac{38}{73}$	**1**	**0**	**0**	**0**	3+12k	**1**	**0**	**1**
$v = \frac{23}{73}$	**1**	**0**	**0**	**0**	3+12k	**0**	**1**	**0**
$u = \frac{20}{73}$	**0**	**3**	**0**	**0**	6+24k	**1**	**1**	**1**
$y = \frac{15}{73}$	**0**	**0**	**4**	**0**	8+36k	**1**	**2**	**0**
$x = \frac{14}{73}$	**0**	**0**	**0**	**1**	1	**0**	**0**	**1**
	3+12k	2+8k	2+9k	1		2+12k	3+12k	1
	8+29k					**6+24k**		

Table D.33. Balance for $FFD = \frac{3}{4} + \frac{29}{24} CSP_R$.

- Assume that neither of the previous two holds. This implies that all items in $\langle 1-3x, \frac{1}{2}]$ have the same size, say w. It further implies that $w+v>1-x$ must hold and thus $w>\frac{2}{3}-x$. We can now tighten the constraint in (D71) by replacing $(1-3x)^+$ by $(\frac{2}{3}-x)^+$. This allows the weight of the item(s) in $\langle 1-3x, \frac{1}{2}]$ to be increased from $\frac{12}{24}$ to $\frac{14}{24}$. This does not affect the optimal solution of the SPP, but it reduces its value to $\frac{7}{24}$. This gives an upper bound for the constant as $c \leqslant \frac{7}{24}+1-\frac{5}{24}=\frac{13}{12}$.

D.2.3.4 $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z \leqslant 3x$ and $v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$; case $\nexists\, y \in \langle \frac{1-x}{4}, \frac{1}{4}]$

We use the following weights and weighting function for the case when there is no bin with largest item $y \in \langle \frac{1-x}{4}, \frac{1}{4}]$,

$$W(z) = V, W(v) = 1 - V \text{ and } W(s) = \begin{cases} \frac{3}{5}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{1}{2}, & s \in \langle 1-3x, \frac{1}{2}] \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1+x}{4}] \\ \frac{4}{15}, & s \in \langle \frac{1}{4}, \frac{1-x}{3}] \\ \frac{1}{5}, & s \in [x, \frac{1}{4}] \end{cases} \tag{D74}$$

This gives the following program.

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{5}a_7 \\ \text{st} & \left(\frac{3-5x}{4}\right)^+ a_1 + \left(\frac{7-19x}{12}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 + \left(\frac{1-x}{3}\right)^+ a_5 + \left(\frac{1}{4}\right)^+ a_6 + xa_7 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \text{ and } 2a_1 + a_4 \leqslant 2 \\ & x \in \langle \frac{2}{11}, \frac{1}{5}] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D75}$$

If $a_1, \ldots, a_6 = 0$ then $\mathcal{R}(V) = 1$, so assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$\tfrac{3-5x}{4}a_1 + \tfrac{7-19x}{12}a_2 + \tfrac{1}{2}a_3 + (1-3x)a_4 + \tfrac{1-x}{3}a_5 + \tfrac{1}{4}a_6 + xa_7 < 1 \tag{D76}$$

Substituting the extremal values of x in this constraint gives the following constraints.

$$\begin{cases} 23a_1 + 13a_2 + 22a_3 + 20a_4 + 12a_5 + 11a_6 + 8a_7 \leqslant 43, & (x = \frac{2}{11}) \\ 30a_1 + 16a_2 + 30a_3 + 24a_4 + 16a_5 + 15a_6 + 12a_7 \leqslant 59, & (x = \frac{1}{5}) \end{cases} \tag{D77}$$

Solving the resulting two IPs for the extremal values of x gives table D.34. From this table we can see that the minimising value for V is $\frac{59}{90}$, which gives a value $\mathcal{R}(\frac{59}{90}) = \frac{107}{90}$. There are exactly two patterns that achieve this value.

Constant The set-packing problem is given in table D.35. Note that pattern 10 cannot represent a valid bin, since we have assumed that there are no bins with largest item in $\langle \frac{1-x}{4}, \frac{1}{4}]$ and the only bins with largest item in $[x, \frac{1-x}{4}]$ are bins with 5 items of size x. So pattern 10 needs to be excluded from the set-packing formulation. With this adjustment the SPP gives a value of $\frac{8}{30}$ for pattern combination 5 and 9. An upper bound for the constant is therefore $c \leqslant \frac{8}{30} + 1 - \frac{1}{5} = \frac{16}{15}$.

Problem Instance Let w be the smallest item in $\langle 1-3x, \frac{1}{2}]$. The requirements on z, v and w imply that both (optimal) patterns cannot be active at the same time; $2v+1 < 2v+w+z < 1+\frac{2}{3}(1-x) \Rightarrow v < \frac{1}{3}(1-x)$. This contradicts with the range that we assumed v to be in. Since there are only two optimal patterns we know that a ratio of $\frac{107}{90}$ is not achievable.

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{59}{90}$	valid for	
$x = \frac{2}{11}$	$[0, \frac{6}{10}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{93}{90}$		
	$[\frac{6}{10}, \frac{19}{30}]$	**0**	**1**	**1**	**0**	**0**	**0**	**1**	$\frac{9}{5} - V$	$\frac{103}{90}$		
	$[\frac{19}{30}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**1**	$\frac{8}{15} + V$	$\rightarrow \frac{107}{90} \leftarrow$		
$x = \frac{1}{5}$	$[0, \frac{1}{2}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{93}{90}$		
	$[\frac{1}{2}, \frac{59}{90}]$	**0**	**2**	**0**	**1**	**0**	**0**	**0**	$\frac{5}{2} - 2V$	$\rightarrow \frac{107}{90} \leftarrow$	$x > \frac{7}{37}$	$2v + w \leqslant 1$
	$[\frac{59}{90}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**1**	$\frac{8}{15} + V$	$\rightarrow \frac{107}{90} \leftarrow$	$x > \frac{1}{7}$	$z < \frac{2}{3}(1 - x)$

Table D.34. Solution of ratio problem (D75)

						*				*	
		1	2	3	4	5	6	7	8	9	10
$W_T \times 30$		29	27	28	26	26	24	28	28	26	24
$(1 - W_T) \times 30$		1	3	2	4	4	6	2	2	4	–
$\langle 1 - 3x, \frac{1}{2}]$	15/30	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1-x}{3}, \frac{1+x}{4}]$	10/30	**0**	**0**	**2**	**2**	**1**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1}{4}, \frac{1-x}{3}]$	8/30	**1**	**0**	**1**	**0**	**2**	**1**	**0**	**2**	**1**	**0**
$[x, \frac{1}{4}]$	6/30	**1**	**2**	**0**	**1**	**0**	**1**	**3**	**2**	**3**	**4**
		1	1	1	1	1	1	1			
			1	1	1	1	1	1			
			1						1	1	
				1	1	1	1	1			
					1		1	1	1	1	

Table D.35. SPP-formulation under weighting function (D74)

D.2.3.5 $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z \leqslant 3x$ and $v \in \langle \frac{7-19x}{12}, \frac{1+x}{4}]$; case $\exists\, y \in \langle \frac{1-x}{4}, \frac{1}{4}]$

We start with the following weights and weighting function for the case when there is a bin with largest item $y \in \langle \frac{1-x}{4}, \frac{1}{4}]$,

$$W(z) = V, W(v) = 1 - V \text{ and } W(s) = \begin{cases} \frac{3}{5}, & s \in \langle \frac{1}{2}, 3x] \\ \frac{1}{2}, & s \in \langle 1-3x, \frac{1}{2}] \\ \frac{1}{3}, & s \in \langle \frac{1-x}{3}, \frac{1+x}{4}] \\ \frac{4}{15}, & s \in \langle \frac{1-y}{3}, \frac{1-x}{3}] \\ \frac{1}{4}, & s \in [y, \frac{1-y}{3}] \\ \frac{1}{5}, & s \in [x, y\rangle \end{cases} \tag{D78}$$

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & \left(\frac{3-5x}{4}\right)^+ a_1 + \left(\frac{7-19x}{12}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 + \left(\frac{1-x}{3}\right)^+ a_5 \\ & \qquad + \left(\frac{1-y}{3}\right)^+ a_6 + ya_7 + xa_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \text{ and } 2a_1 + a_4 \leqslant 2 \\ & x \in \langle \frac{2}{11}, \frac{1}{5}], \; y \in \langle \frac{1-x}{4}, \frac{1}{4}] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D79}$$

If $a_1, \ldots, a_6 = 0$ then $\mathcal{R}(V)$ reduces to the following two-variable problem.

$$\mathcal{R}_0 = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & ya_7 + xa_8 \leqslant 1 \\ & x \in \langle \frac{2}{11}, \frac{1}{5}], \; y \in \langle \frac{1-x}{4}, \frac{1}{4}] \\ & a_i \in \mathbb{N} \end{array}} = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{20}(5a_7 + 4a_8) \\ \text{st} & 9a_7 + 8a_8 \leqslant 43 \\ & a_i \in \mathbb{N} \end{array}} \tag{D80}$$

If $a_7 = 0$ then $\mathcal{R}_0 = 1$, so assume that $a_7 \geqslant 1$ and we can replace the constraint by $\frac{1-x}{4}a_7 + xa_8 < 1$. Substituting the extremal values of x gives the following constraints.

$$\begin{cases} 9a_7 + 8a_8 \leqslant 43, & (x = \frac{2}{11}) \\ \;\; a_7 + \;\; a_8 \leqslant \;\; 4, & (x = \frac{1}{5}) \end{cases} \tag{D81}$$

It is easy to verify that the constraint for $x = \frac{1}{5}$ is implied by the constraint for $x = \frac{2}{11}$. So we end up with the program on the right hand side of (D80). Simple enumeration will show that this problem is maximised for $[a_7, a_8] = [3, 2]$ with value $\frac{23}{20} = 1.15$. So that,

$$a_1, \ldots, a_6 = 0 \;\Rightarrow\; \mathcal{R}(V) = \tfrac{23}{20} \tag{D82}$$

		V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\mathcal{R}(V)$	$V^* = \frac{23}{36}$	valid for
$y = \frac{1-x}{4}$	$x = \frac{2}{11}$	$[0, \frac{11}{20}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{195}{180}$	
		$[\frac{11}{20}, \frac{37}{60}]$	**0**	**2**	**0**	**0**	**0**	**0**	**1**	**1**	$\frac{49}{20} - 2V$	$\frac{211}{180}$	
		$[\frac{37}{60}, \frac{13}{20}]$	**0**	**1**	**0**	**0**	**1**	**0**	**2**	**0**	$\frac{11}{6} - V$	$\frac{215}{180}$	
		$[\frac{13}{20}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	$\frac{8}{15} + V$	$\frac{211}{180}$	
	$x = \frac{1}{5}$	$[0, \frac{1}{2}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{195}{180}$	
		$[\frac{1}{2}, \frac{23}{36}]$	**0**	**2**	**0**	**0**	**0**	**0**	**2**	**0**	$\frac{5}{2} - 2V$	$\rightarrow \frac{220}{180} \leftarrow$	$x > \frac{2}{11}$
		$[\frac{23}{36}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**0**	$\frac{7}{12} + V$	$\rightarrow \frac{220}{180} \leftarrow$	$x > \frac{2}{11}$
$y = \frac{1}{4}$	$x = \frac{2}{11}$	$[0, \frac{6}{10}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{195}{180}$	
		$[\frac{6}{10}, \frac{19}{30}]$	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	$\frac{9}{5} - V$	$\frac{209}{180}$	
		$[\frac{19}{30}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	$\frac{8}{15} + V$	$\frac{211}{180}$	
	$x = \frac{1}{5}$	$[0, \frac{1}{2}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{195}{180}$	
		$[\frac{1}{2}, \frac{59}{90}]$	**0**	**2**	**0**	**1**	**0**	**0**	**0**	**0**	$\frac{5}{2} - 2V$	$\rightarrow \frac{220}{180} \leftarrow$	$x > \frac{7}{37}$
		$[\frac{59}{90}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	$\frac{8}{15} + V$	$\frac{211}{180}$	

Table D.36. Solution of ratio problem (D79)

Now assume that at least one of $a_1, \ldots, a_6$ is non-zero. This means that we can replace the first constraint in program (D79) by

$$\tfrac{3-5x}{4}a_1 + \tfrac{7-19x}{12}a_2 + \tfrac{1}{2}a_3 + (1-3x)a_4 + \tfrac{1-x}{3}a_5 + \tfrac{1-y}{3}a_6 + ya_7 + xa_8 < 1 \qquad \text{(D83)}$$

Substituting the extremal values of y, gives the following constraints.

$$\begin{cases} \tfrac{3-5x}{4}a_1 + \tfrac{7-19x}{12}a_2 + \tfrac{1}{2}a_3 + (1-3x)a_4 + \tfrac{1-x}{3}a_5 + \tfrac{3+x}{12}a_6 + \tfrac{1-x}{4}a_7 + xa_8 < 1, & (y = \tfrac{1-x}{4}) \\ \tfrac{3-5x}{4}a_1 + \tfrac{7-19x}{12}a_2 + \tfrac{1}{2}a_3 + (1-3x)a_4 + \tfrac{1-x}{3}a_5 + \tfrac{1}{4}a_6 + \tfrac{1}{4}a_7 + xa_8 < 1, & (y = \tfrac{1}{4}) \end{cases} \qquad \text{(D84)}$$

Substituting the extremal values of x in each of these constraints gives the following.

$$\begin{cases} 69a_1 + 39a_2 + 66a_3 + 60a_4 + 36a_5 + 35a_6 + 27a_7 + 24a_8 \leqslant 131, & (y = \tfrac{1-x}{4}) \text{ and } (x = \tfrac{2}{11}) \\ 30a_1 + 16a_2 + 30a_3 + 24a_4 + 16a_5 + 16a_6 + 12a_7 + 12a_8 \leqslant 59, & (y = \tfrac{1-x}{4}) \text{ and } (x = \tfrac{1}{5}) \\ 69a_1 + 39a_2 + 66a_3 + 60a_4 + 36a_5 + 33a_6 + 33a_7 + 24a_8 \leqslant 131, & (y = \tfrac{1}{4}) \text{ and } (x = \tfrac{2}{11}) \\ 30a_1 + 16a_2 + 30a_3 + 24a_4 + 16a_5 + 15a_6 + 15a_7 + 12a_8 \leqslant 59, & (y = \tfrac{1}{4}) \text{ and } (x = \tfrac{1}{5}) \end{cases} \qquad \text{(D85)}$$

Solving the resulting four IPs for the extremal values of y and x gives table D.36. The minimising value of V is $\frac{23}{36}$, which gives a value of $\frac{11}{9}$. There are three patterns that achieve this value.

Weighting Function We now strengthen the weighting function.

The optimal patterns indicate that we can increase the weight of items corresponding to $\left(\frac{1}{2}\right)^+$, $\left(\frac{1-y}{3}\right)^+$ and x, without increasing the ratio. We fix $V = \frac{23}{36}$ and consider the following problem.

$$\mathcal{R}(W) = \begin{array}{|ll|} \hline \text{Max} & \frac{23}{36}a_1 + \frac{13}{36}a_2 + (1-2W)a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{1-W}{3}a_6 + \frac{1}{4}a_7 + Wa_8 \\ \text{st} & \left(\frac{3-5x}{4}\right)^+ a_1 + \left(\frac{7-19x}{12}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 + \left(\frac{1-x}{3}\right)^+ a_5 \\ & \qquad + \left(\frac{1-y}{3}\right)^+ a_6 + ya_7 + xa_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \text{ and } 2a_1 + a_4 \leqslant 2 \\ & x \in \left\langle \frac{2}{11}, \frac{1}{5}\right], \; y \in \left\langle \frac{1-x}{4}, \frac{1}{4}\right] \text{ and } a_i \in \mathbb{N} \\ \hline \end{array} \tag{D86}$$

The objective is to find the maximum value of W, while not exceeding the ratio of $\frac{11}{9}$. First condition on $a_1, \ldots, a_6 = 0$. This gives the following two-variable problem, see (D80).

$$\mathcal{R}(W) = \begin{array}{|ll|} \hline \text{Max} & \frac{1}{4}a_7 + Wa_8 \\ \text{st} & 9a_7 + 8a_8 \leqslant 43 \\ & a_i \in \mathbb{N} \\ \hline \end{array} \tag{D87}$$

By simple enumeration of the feasible patterns we can determine the maximum value of W, such that $\mathcal{R}(W)$ does not exceed $\frac{11}{9}$. This value is determined by the pattern $[a_7, a_8] = [3, 2]$ as $W \leqslant \frac{17}{72}$. We now fix W to $\frac{17}{72}$ and determine the maximum weights for $\left(\frac{1}{2}\right)^+$ and $\left(\frac{1-y}{3}\right)^+$. These are $\frac{11}{18}$ and $\frac{19}{72}$ respectively. This finally gives weighting function (8.69).

Constant The set-packing problem is given in table D.37. Note that since we have conditioned on the presence of a bin with largest item $y \in \left\langle \frac{1-x}{4}, \frac{1}{4}\right]$, it follows that patterns 1, 2, 5, 8, 10, 11 and 15 cannot represent a valid bin. For example, consider pattern 5; after placing the first two items in the bin it is filled to a level $\leqslant 2 \times \frac{1+x}{4}$ which is $\leqslant 1 - y$, as is easily verified, and FFD would have placed the item y in a bin represented by this pattern. Further, since the only bin with largest item in $[x, y\rangle$ is the singleton bin, pattern 15 cannot represent a transition bin. With this adjustment the SPP gives a value of $\frac{15}{72}$ for pattern combination (9, 14). An upper bound for the constant is therefore $c \leqslant \frac{15}{72} + 1 - \frac{17}{72} = \frac{35}{36}$.

Problem Instance A balance with sizes is given in table D.38

										*					*	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$W_T \times 72$		71	70	67	66	65	62	61	60	60	71	70	71	70	69	68
$(1 - W_T) \times 72$		–	–	5	6	–	10	11	–	12	–	–	1	2	3	–
$\langle 1-3x, \frac{1}{2}]$	36/72	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1-x}{3}, \frac{1+x}{4}]$	24/72	**0**	**0**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1-y}{3}, \frac{1-x}{3}]$	19/72	**0**	**0**	**1**	**0**	**0**	**2**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
$[y, \frac{1-y}{3}]$	18/72	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**2**	**1**	**0**	**3**	**2**	**1**	**0**
$[x, y\rangle$	17/72	**1**	**2**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**2**	**3**	**1**	**2**	**3**	**4**
				1	1	1	1	1	1	1						
													1	1	1	
						1			1				1	1	1	

Table D.37. SPP-formulation under weighting function (8.69)

	FFD-bins							LP-patterns				
$z = \frac{72}{140}$	**1**	**0**	**0**	**0**	**0**	**0**	10+6k	**1**	**1**	**1**	**1**	**0**
$v = \frac{41}{140}$	**1**	**0**	**0**	**0**	**0**	**0**	10+6k	**0**	**0**	**0**	**0**	**2**
$u = \frac{39}{140}$	**0**	**3**	**1**	**0**	**0**	**0**	7+6k	**1**	**1**	**0**	**0**	**0**
$y = \frac{37}{140}$	**0**	**0**	**2**	**0**	**0**	**0**	2	**0**	**0**	**1**	**0**	**0**
$y = \frac{29}{140}$	**0**	**0**	**0**	**4**	**1**	**0**	17+12k	**1**	**0**	**0**	**2**	**2**
$x = \frac{28}{140}$	**0**	**0**	**0**	**0**	**3**	**1**	4	**0**	**1**	**1**	**0**	**0**
	10+6k	2+2k	1	4+3k	1	1		5+6k	2	2	1	5+3k
	19+11k							**15+9k**				

Table D.38. Balance for $FFD = \frac{2}{3} + \frac{11}{9}OPT$

D.2.3.6 $x \in \left\langle \frac{2}{11}, \frac{1}{5} \right]$, $z \leqslant 3x$ and $v \in \left\langle \frac{1-x}{3}, \frac{7-19x}{12} \right]$

Ratio Problem Substituting the lower bounds for z and v in program (D70), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \begin{array}{|ll|}\hline \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{1}{3}a_5 + \frac{4}{15}a_6 + \frac{1}{4}a_7 + \frac{1}{5}a_8 \\ \text{st} & \left(\frac{5+7x}{12}\right)^+ a_1 + \left(\frac{1-x}{3}\right)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 + \left(\frac{1-x}{3}\right)^+ a_5 \\ & \qquad + \left(\frac{1}{4}\right)^+ a_6 + \left(\frac{1-x}{4}\right)^+ a_7 + x a_8 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \\ & x \in \left\langle \frac{2}{11}, \frac{1}{5} \right] \text{ and } a_i \in \mathbb{N} \\ \hline \end{array} \tag{D88}$$

If $a_1, \ldots, a_7 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$\tfrac{5+7x}{12}a_1 + \tfrac{1-x}{3}a_2 + \tfrac{1}{2}a_3 + \big(1-3x\big)a_4 + \tfrac{1-x}{3}a_5 + \tfrac{1}{4}a_6 + \tfrac{1-x}{4}a_7 + xa_8 < 1 \tag{D89}$$

Substituting the extremal values of x, gives the following constraints.

$$\begin{cases} 23a_1 + 12a_2 + 22a_3 + 20a_4 + 12a_5 + 11a_6 + 9a_7 + 8a_8 \leqslant 43, & (x = \frac{2}{11}) \\ 32a_1 + 16a_2 + 30a_3 + 24a_4 + 16a_5 + 15a_6 + 12a_7 + 12a_8 \leqslant 59, & (x = \frac{1}{5}) \end{cases} \tag{D90}$$

Solving the resulting IP for the extremal values of x gives table D.39. From this table we can deduce that the minimising value for V is $\frac{119}{180}$, which gives a value $\mathcal{R}(\frac{119}{180}) = \frac{43}{36}$. There are two patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of items corresponding to $\left(\frac{1}{2}\right)^+$ and $\left(1-3x\right)^+$ to $\frac{109}{180}$ and $\frac{31}{60}$ respectively. However, this does not reduce the constant derived from the set-packing formulation.

Constant The set-packing problem is given in table D.40. It has a value of $\frac{19}{60}$ for pattern combination $(7, 13, 18)$. An upper bound for the constant therefore is $c \leqslant \frac{19}{60} + 1 - \frac{1}{5} = \frac{67}{60}$.

Problem Instance No instance that achieves the asymptotic ratio of $\frac{43}{36}$ was found.

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\mathcal{R}(V)$	$V^* = \frac{119}{180}$	valid for	
$x = \frac{2}{11}$	$[0, \frac{1}{2}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{183}{180}$		
	$[\frac{1}{2}, \frac{13}{20}]$	**0**	**2**	**0**	**0**	**0**	**0**	**2**	**0**	$\frac{5}{2} - 2V$	$\frac{212}{180}$		
	$[\frac{13}{20}, \frac{79}{120}]$	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	$\frac{37}{20} - V$	$\frac{214}{180}$		
	$[\frac{79}{120}, 1]$	**1**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	$\frac{8}{15} + V$	$\rightarrow \frac{215}{180} \leftarrow$	$x < \frac{1}{5}$	$v > \frac{1-x}{3}$
$x = \frac{1}{5}$	$[0, \frac{29}{60}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{183}{180}$		
	$[\frac{29}{60}, \frac{2}{3}]$	**0**	**2**	**0**	**0**	**0**	**1**	**1**	**0**	$\frac{151}{60} - 2V$	$\rightarrow \frac{215}{180} \leftarrow$	$x > \frac{2}{11}$	$v < \frac{2+x}{8}$
	$[\frac{2}{3}, 1]$	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	$\frac{31}{60} + V$	$\frac{212}{180}$		

Table D.39. Solution of ratio problem (D88)

									*					*					*
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$W_T \times 60$		58	57	54	56	55	52	52	51	59	56	59	56	58	55	52	57	54	51
$(1 - W_T) \times 60$		2	3	6	4	5	8	8	9	1	4	1	4	2	5	8	3	6	9
$\langle 1 - 3x, \frac{1}{2}]$	30/60	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1-x}{3}, \frac{7-19x}{12}]$	20/60	**0**	**0**	**0**	**2**	**2**	**2**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
$\langle \frac{1}{4}, \frac{1-x}{3}]$	16/60	**1**	**0**	**0**	**1**	**0**	**0**	**2**	**1**	**0**	**0**	**2**	**2**	**1**	**1**	**1**	**0**	**0**	**0**
$\langle \frac{1-x}{4}, \frac{1}{4}]$	15/60	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**2**	**1**	**0**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	12/60	**1**	**1**	**2**	**0**	**0**	**1**	**0**	**0**	**2**	**3**	**1**	**2**	**1**	**2**	**3**	**1**	**2**	**3**
		1	1	1	1	1	1	1	1	1	1								
			1	1								1	1	1	1	1			
				1													1	1	1
						1	1		1	1	1	1	1	1	1	1			
							1			1	1						1	1	1
													1		1	1	1	1	1

Table D.40. SPP-formulation under weighting function (8.71)

D.2.3.7 $x \in \langle \frac{2}{11}, \frac{1}{5}]$, $z > 3x$ and $v \in \langle 1 - 4x, \frac{1-x}{3}]$

Ratio Problem Substituting the lower bounds for z and v in program (D70), and deleting all other constraints in z and v gives the following program.

$$\mathcal{R}(V) = \begin{array}{|ll|} \text{Max} & Va_1 + (1-V)a_2 + \frac{3}{5}a_3 + \frac{1}{2}a_4 + \frac{4}{15}a_5 + \frac{1}{4}a_6 + \frac{1}{5}a_7 \\ \text{st} & \frac{2}{3}(1-x)^+ a_1 + (1-4x)^+ a_2 + \left(\frac{1}{2}\right)^+ a_3 + (1-3x)^+ a_4 + \left(\frac{1}{4}\right)^+ a_5 \\ & \qquad + \left(\frac{1-x}{4}\right)^+ a_6 + xa_7 \leqslant 1 \\ & 5a_1 + a_2 \leqslant 5 \\ & 2a_1 + a_4 \leqslant 2 \\ & x \in \langle \frac{2}{11}, \frac{1}{5}] \text{ and } a_i \in \mathbb{N} \end{array} \tag{D91}$$

If $a_1, \ldots, a_6 = 0$ then $\mathcal{R}(V) = 1$. We may therefore assume that at least one of them is non-zero. This means that we can replace the first constraint by

$$\frac{2}{3}(1-x)a_1 + (1-4x)a_2 + \frac{1}{2}a_3 + \big(1-3x\big)a_4 + \frac{1}{4}a_5 + \frac{1-x}{4}a_6 + xa_7 < 1 \tag{D92}$$

Substituting the extremal values of x, gives the following constraints.

$$\begin{cases} 24a_1 + 12a_2 + 22a_3 + 20a_4 + 11a_5 + 9a_6 + 8a_7 \leqslant 43, & (x = \frac{2}{11}) \\ 32a_1 + 12a_2 + 30a_3 + 24a_4 + 15a_5 + 12a_6 + 12a_7 \leqslant 59, & (x = \frac{1}{5}) \end{cases} \tag{D93}$$

Solving the resulting IP for the extremal values of x gives table D.41. From table D.41 we can see that the minimising value for V is $\frac{209}{300}$, which gives a value $\mathcal{R}(\frac{209}{300}) = \frac{91}{75}$. There are two [optimal] patterns that achieve this value.

Weighting Function The optimal patterns indicate that we can increase the weight of items corresponding to $\left(\frac{1}{2}\right)^+$, $\big(1-3x\big)^+$ and x without increasing the ratio. These can be increased to $\frac{91}{150}$, $\frac{91}{150}$ and $\frac{139}{600}$ respectively. This gives the weighting function (8.73).

Constant The set-packing problem, under this (stronger) weighting function, is given in table D.42. It has a value of $\frac{34}{600}$ for pattern combination $(2, 7)$. An upper bound for the constant therefore is $c \leqslant \frac{34}{600} + 1 - \frac{139}{600} = \frac{33}{40}$.

Problem Instance There is no instance that achieves the asymptotic ratio of $\frac{91}{75}$.

Comment The requirements on v imply that both patterns cannot be active at the same time; $\frac{2-5x}{4} < v \leqslant \frac{1}{4} \Rightarrow x > \frac{1}{5}$. Since there are only two optimal patterns we know that the ratio of $\frac{91}{75}$ is not achievable. Moreover, using $V = \frac{95}{136}$ and the generic weighting function with $W = \frac{31}{136}$ one can show that the ratio is upper bounded by $\frac{41}{34}$. However, the bound resulting from $r \leqslant \frac{91}{75}$ is sufficient for our immediate purposes.

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{209}{300}$	valid for	
$x = \frac{2}{11}$	$[0, \frac{1}{2}]$	**0**	**3**	**0**	**0**	**0**	**0**	**0**	$3 - 3V$	$\frac{273}{300}$		
	$[\frac{1}{2}, \frac{13}{20}]$	**0**	**2**	**0**	**0**	**0**	**2**	**0**	$\frac{5}{2} - 2V$	$\frac{332}{300}$		
	$[\frac{13}{20}, \frac{27}{40}]$	**0**	**1**	**1**	**0**	**0**	**1**	**0**	$\frac{37}{20} - V$	$\frac{346}{300}$		
	$[\frac{27}{40}, 1]$	**1**	**0**	**0**	**0**	**0**	**2**	**0**	$\frac{1}{2} + V$	$\frac{359}{300}$		
$x = \frac{1}{5}$	$[0, \frac{209}{300}]$	**0**	**4**	**0**	**0**	**0**	**0**	**0**	$4 - 4V$	$\rightarrow \frac{364}{300} \leftarrow$	$x > \frac{3}{16}$	$v \leqslant \frac{1}{4}$
	$[\frac{209}{300}, 1]$	**1**	**0**	**0**	**0**	**1**	**1**	**0**	$\frac{31}{60} + V$	$\rightarrow \frac{364}{300} \leftarrow$	$x > \frac{2}{11}$	$v > \frac{2-5x}{4}$

Table D.41. Solution of ratio problem (D91)

		1	2 *	3	4	5	6	7 *
$W_T \times 600$		598	599	588	577	589	578	567
$(1 - W_T) \times 600$		2	1	12	23	11	22	33
$\langle\frac{1}{4}, \frac{1-x}{3}]$	160/600	**2**	**1**	**1**	**1**	**0**	**0**	**0**
$\langle\frac{1-x}{4}, \frac{1}{4}]$	150/600	**0**	**2**	**1**	**0**	**3**	**2**	**1**
$[x, \frac{1-x}{4}]$	139/600	**2**	**1**	**2**	**3**	**1**	**2**	**3**
		1	1	1	1			
						1	1	1
		1		1	1	1	1	1

Table D.42. SPP-formulation under weighting function (8.73)

D.2.4 Further assumptions

We can now use bound (8.70) and weighting function (8.69) to draw further conclusions on the minimal configuration, with respect to CSP_R. To do this we use the following. Suppose that we have a bin with weight $W_{\text{bin}} > 1$ and this bin occurs n times. This gives

$$FFD \leqslant \sum_{j|W_j<1} (1 - W_j) \; + \; (1 - W_{\text{bin}})\,n \; + \; \tfrac{11}{9} CSP_R \tag{D94}$$

and an upper bound for the constant is $\frac{35}{36} - \big(1 - W_{\text{bin}}\big)n$. But we already have an example that achieves $FFD = \frac{22}{27} + \frac{11}{9} CSP_R$. This has consequences for the minimal configuration.

D2 Assumption *There is a bin with largest item* $u \in \langle \frac{1-x}{3}, v]$

Proof. *If there is no such bin then patterns 3–9 in table D.37 need to be excluded from the SPP-formulation. This gives an upper bound for the constant of* $\frac{3}{72} + 1 - \frac{17}{72} = \frac{29}{36} = 0.80555...$ □

D3 Assumption *Block 2 (in diagram 8.44) is empty*

Proof. *This follows from lemma 8.29. There are two possibilities for a bin in block 2.*

- *It contains two items, with the smallest one* $> 1 - 3x$. $W_{\text{bin}} = \frac{11}{18} + \frac{43}{72} \;\Rightarrow\; c \leqslant \frac{55}{72} = 0.76388...$
- *It contains three items. Since there is a bin with largest item* $u \in \langle \frac{1-x}{3}, v]$, *by assumption D2, it must be that the second item placed in the bin is larger than or equal to* u. $W_{\text{bin}} \geqslant \frac{11}{18} + \frac{1}{3} + \frac{17}{72} \;\Rightarrow\; c \leqslant \frac{19}{24} = 0.79166...$

□

D4 Assumption *There are no bins with largest item in* $\langle 1 - 3x, \frac{1}{2}]$

Proof. *There are the following possibilities for a bin with largest item in* $\langle 1 - 3x, \frac{1}{2}]$.

- *The bin contains two items in* $\langle 1 - 3x, \frac{1}{2}]$. $W_{\text{bin}} = 2 \times \frac{43}{72} \;\Rightarrow\; c \leqslant \frac{7}{9} = 0.77777...$
- *The bin contains one item in* $\langle 1 - 3x, \frac{1}{2}]$ *and two other items. Since there is a bin with largest item* u, *it must be that the second item placed in the bin is larger than or equal to* u. $W_{\text{bin}} \geqslant \frac{43}{72} + \frac{1}{3} + \frac{17}{72} \;\Rightarrow\; c \leqslant \frac{29}{36} = 0.80555...$
- *If there is only one additional item, say* u_1 *then* $u_1 \leqslant 1 - 3x$ *and since* v *is the largest item* $\leqslant 1 - 3x$ $u_1 \leqslant v$ *must hold. This implies that the bin is filled to a level* $w + u_1 \leqslant \frac{1}{2} + v \leqslant \frac{1}{2} + \frac{1+x}{4} \leqslant 1 - x$. *So FFD would have placed another item in this bin, which leads to a contradiction.* □

D5 Assumption *There are no bins with 5 items of size* x.

Proof. $W_{\text{bin}} = 5 \times \frac{17}{72} \;\Rightarrow\; c \leqslant \frac{57}{72} = 0.79166...$ □

D6 Assumption *There are at most 5 bins with largest item in* $\langle \frac{1-y}{3}, \frac{1-x}{3}]$.

Proof. *The weight of such bin is at least* $3 \times \frac{19}{72} + \frac{17}{72} = \frac{37}{36}$. *So that* $n < 36 \times \frac{17}{108}$ *and thus* $n \leqslant 5$. □

Combining these assumptions we can conclude that the minimal configuration, with respect to CSP_R, must be of the form as shown in diagram D.12. [1]

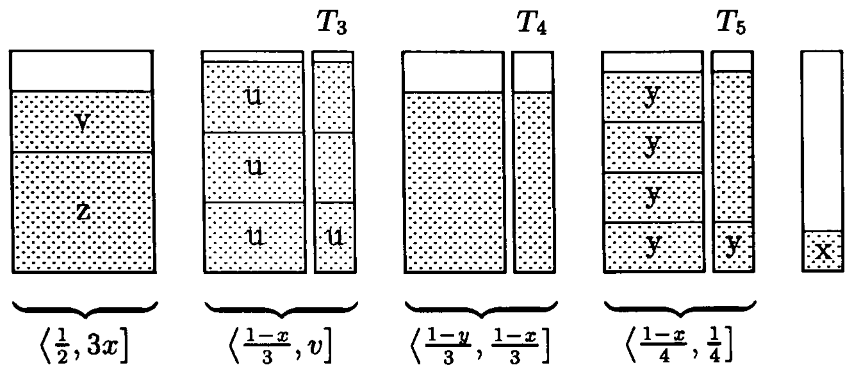


Diagram D.12. Minimal configuration for $\beta(x) = 5$

[1]Note that the optimal patterns in table D.36 imply that $2v + 2y \leqslant 1$ and $z + u + y \leqslant 1$ are necessary conditions to achieve a ratio of 11/9.

D.3 Lists with no 1-items

D.3.1 $\alpha = 2$ and $\beta = 6$, and there is a bin with largest item $y \in \left\langle \frac{1-x}{5}, \frac{1}{5} \right]$

We start with a variant of the generic weighting function, to end up with the weighting function on the RHS.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-y}{2}, \frac{1}{2} \right] \\ \frac{1-V}{2}, & s \in \left\langle \frac{1}{3}, \frac{1-y}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-y}{3}, \frac{1}{3} \right] \\ \frac{1-V}{3}, & s \in \left\langle \frac{1}{4}, \frac{1-y}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-y}{4}, \frac{1}{4} \right] \\ V, & s \in \left[y, \frac{1-y}{4} \right] \\ \frac{1}{6}, & s \in [x, y\rangle \end{cases} \xrightarrow[V=1/5]{\text{Minimising value}} W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-y}{2}, \frac{1}{2} \right] \\ \frac{2}{5}, & s \in \left\langle \frac{1}{3}, \frac{1-y}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-y}{3}, \frac{1}{3} \right] \\ \frac{4}{15}, & s \in \left\langle \frac{1}{4}, \frac{1-y}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-y}{4}, \frac{1}{4} \right] \\ \frac{1}{5}, & s \in \left[y, \frac{1-y}{4} \right] \\ \frac{1}{6}, & s \in [x, y\rangle \end{cases} \tag{D95}$$

The maximum pattern-weight is given by the following program,

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + \frac{1-V}{3}a_4 + \frac{1}{4}a_5 + Va_6 + \frac{1}{6}a_7 \\ \text{st} & \left(\frac{1-y}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-y}{3}\right)^+ a_3 + \left(\frac{1}{4}\right)^+ a_4 + \left(\frac{1-y}{4}\right)^+ a_5 + ya_6 + xa_7 \leqslant 1 \\ & x \in \left\langle \frac{1}{7}, \frac{1}{6} \right],\ y \in \left\langle \frac{1-x}{5}, \frac{1}{5} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D96}$$

We replace x by $(1/7)^+$ and solve the two programs for the extremal values of y. This gives table D.43 and the minimising value for V of $1/5$, which gives $\mathcal{R}(1/5) = 71/60$.

D.3.2 $\alpha = 2$ and $\beta = 6$, and there is no bin with largest item $y \in \left\langle \frac{1-x}{5}, \frac{1}{5} \right]$

We now split the range $[x, 1/2]$ according to x and use the following variant of the generic weighting function.

$$W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{1-V}{2}, & s \in \left\langle \frac{1}{3}, \frac{1-x}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right] \\ \frac{1-V}{3}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{1-V}{4}, & s \in \left\langle \frac{1}{5}, \frac{1-x}{4} \right] \\ V, & s \in \left[x, \frac{1}{5} \right] \end{cases} \xrightarrow[V=1/6]{\text{Minimising value}} W(s) = \begin{cases} \frac{1}{2}, & s \in \left\langle \frac{1-x}{2}, \frac{1}{2} \right] \\ \frac{5}{12}, & s \in \left\langle \frac{1}{3}, \frac{1-x}{2} \right] \\ \frac{1}{3}, & s \in \left\langle \frac{1-x}{3}, \frac{1}{3} \right] \\ \frac{5}{18}, & s \in \left\langle \frac{1}{4}, \frac{1-x}{3} \right] \\ \frac{1}{4}, & s \in \left\langle \frac{1-x}{4}, \frac{1}{4} \right] \\ \frac{5}{24}, & s \in \left\langle \frac{1}{5}, \frac{1-x}{4} \right] \\ \frac{1}{6}, & s \in \left[x, \frac{1}{5} \right] \end{cases} \tag{D97}$$

The maximum pattern-weight is given by the following program,

$$\mathcal{R}(V) = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{2}a_1 + \frac{1-V}{2}a_2 + \frac{1}{3}a_3 + \frac{1-V}{3}a_4 + \frac{1}{4}a_5 + \frac{1-V}{4}a_6 + Va_7 \\ \text{st} & \left(\frac{1-x}{2}\right)^+ a_1 + \left(\frac{1}{3}\right)^+ a_2 + \left(\frac{1-x}{3}\right)^+ a_3 + \left(\frac{1}{4}\right)^+ a_4 + \left(\frac{1-x}{4}\right)^+ a_5 + \left(\frac{1}{5}\right)^+ a_6 + xa_7 \leqslant 1 \\ & x \in \left\langle \frac{1}{7}, \frac{1}{6} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D98}$$

Solving the resulting programs for the extremal values of x gives table D.44, and a minimising value for V of $1/6$, which gives $\mathcal{R}(1/6) = 7/6$.

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{1}{5}$
$y = \frac{1}{6}$	$[0, \frac{1}{6}]$	**0**	**2**	**1**	**0**	**0**	**0**	**0**	$\frac{4}{3} - V$	$\frac{68}{60}$
	$[\frac{1}{6}, \frac{1}{5}]$	**0**	**1**	**1**	**0**	**1**	**1**	**0**	$\frac{13}{12} + \frac{1}{2}V$	$\rightarrow \frac{71}{60} \leftarrow$
	$[\frac{1}{5}, \frac{5}{24}]$	**0**	**0**	**1**	**0**	**1**	**3**	**0**	$\frac{7}{12} + 3V$	$\rightarrow \frac{71}{60} \leftarrow$
	$[\frac{5}{24}, 1]$	**0**	**0**	**0**	**0**	**0**	**5**	**1**	$\frac{1}{6} + 5V$	$\frac{70}{60}$
$y = \frac{1}{5}$	$[0, \frac{1}{6}]$	**1**	**1**	**0**	**1**	**0**	**0**	**0**	$\frac{4}{3} - \frac{5}{6}V$	$\frac{70}{60}$
	$[\frac{1}{6}, \frac{1}{4}]$	**1**	**0**	**0**	**1**	**1**	**0**	**1**	$\frac{5}{4} - \frac{1}{3}V$	$\rightarrow \frac{71}{60} \leftarrow$
	$[\frac{1}{4}, 1]$	**0**	**0**	**0**	**0**	**0**	**4**	**1**	$\frac{1}{6} + 4V$	$\frac{58}{60}$

Table D.43. Solution of ratio problem (D96)

	V-range	a_1	a_2	a_3	a_4	a_5	a_6	a_7	$\mathcal{R}(V)$	$V^* = \frac{1}{6}$
$x = \frac{1}{7}$	$[0, \frac{1}{6}]$	**0**	**2**	**1**	**0**	**0**	**0**	**0**	$\frac{4}{3} - V$	$\rightarrow \frac{42}{36} \leftarrow$
	$[\frac{1}{6}, \frac{5}{26}]$	**0**	**1**	**0**	**0**	**1**	**0**	**3**	$\frac{3}{4} + \frac{5}{2}V$	$\rightarrow \frac{42}{36} \leftarrow$
	$[\frac{5}{26}, \frac{1}{4}]$	**0**	**0**	**0**	**1**	**0**	**0**	**5**	$\frac{1}{3} + \frac{14}{3}V$	$\frac{40}{36}$
	$[\frac{1}{4}, 1]$	**0**	**0**	**0**	**0**	**0**	**0**	**6**	$6V$	$\frac{36}{36}$
$x = \frac{1}{6}$	$[0, \frac{1}{6}]$	**0**	**2**	**1**	**0**	**0**	**0**	**0**	$\frac{4}{3} - V$	$\rightarrow \frac{42}{36} \leftarrow$
	$[\frac{1}{6}, \frac{1}{5}]$	**0**	**1**	**1**	**0**	**1**	**0**	**1**	$\frac{13}{12} + \frac{1}{2}V$	$\rightarrow \frac{42}{36} \leftarrow$
	$[\frac{1}{5}, \frac{1}{4}]$	**0**	**0**	**1**	**0**	**1**	**0**	**3**	$\frac{7}{12} + 3V$	$\frac{39}{36}$
	$[\frac{1}{4}, \frac{1}{3}]$	**0**	**0**	**1**	**0**	**0**	**0**	**4**	$\frac{1}{3} + 4V$	$\frac{36}{36}$
	$[\frac{1}{3}, 1]$	**0**	**0**	**0**	**0**	**0**	**0**	**5**	$5V$	$\frac{30}{36}$

Table D.44. Solution of ratio problem (D98)

D.3.3 $\alpha \geqslant 3$, $\beta = \alpha + 2$ and there is a bin with largest item $y \in \left\langle \frac{1-x}{\alpha+1}, \frac{1}{\alpha+1} \right]$

The maximum pattern-weight, under weighting function (8.87), is given by the following program:

$$\mathcal{R} \;=\; \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \left(\frac{1-x}{\alpha}\right)^+ a_1 + \left(\frac{1-y}{\alpha}\right)^+ a_2 + y a_3 + x a_4 \leqslant 1 \\ & x \in \left\langle \frac{1}{\alpha+3}, \frac{1}{\alpha+2} \right],\; y \in \left\langle \frac{1-x}{\alpha+1}, \frac{1}{\alpha+1} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D99}$$

One can show that this program is basically the ‘FFD subset-sum problem’; program (B24) on page 171, and that $\mathcal{R} = \mathcal{S}_\alpha(\alpha+2) = \frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$. However, for our purposes it is sufficient to prove the bound $\mathcal{R} \leqslant \mathcal{S}_\alpha(\alpha+2)$. We do this by [first] conditioning on the values of a_1 and a_2.

D.3.3.1 $a_1 = a_2 = 0$

When neither a_1 nor a_2 is active, then (D99) reduces to the following two-variable problem.

$$\mathcal{R}_0 \;=\; \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \left(\frac{1-x}{\alpha+1}\right)^+ a_3 + x a_4 \leqslant 1 \\ & x \in \left\langle \frac{1}{\alpha+3}, \frac{1}{\alpha+2} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \;=\; \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 < \frac{\alpha+3}{\alpha+2} \\ & a_i \in \mathbb{N} \end{array}} \tag{D100}$$

If, additionally $a_3 = 0$, then $\mathcal{R}_0 = 1$ so assume $a_3 \geqslant 1$ and we can replace the constraint (after scaling) by $\frac{1}{\alpha+1}a_3 + \frac{x}{1-x}a_4 < \frac{1}{1-x}$. Substituting the extremal values of x gives the following constraints.

$$\begin{cases} \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 < \frac{\alpha+3}{\alpha+2}, & (x = \frac{1}{\alpha+3}) \\ \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1}, & (x = \frac{1}{\alpha+2}) \end{cases} \tag{D101}$$

The constraint for $x = \frac{1}{\alpha+2}$ leads to a one-variable problem in a_3, since a_4 is dominated by a_3. This problem is easily solved and has value 1. The constraint for $x = \frac{1}{\alpha+3}$ leads to the subset-sum problem on the RHS of (D100). This is maximised for $[a_3, a_4] = [\alpha, 2]$ with value $\frac{\alpha+3}{\alpha+2} - \frac{1}{(\alpha+1)(\alpha+2)}$.
This gives $\mathcal{R}_0 = 1 + \frac{\alpha}{(\alpha+1)(\alpha+2)}$, and it is easily shown that $\mathcal{R}_0 < \mathcal{S}_\alpha(\alpha+2)$ holds for $\alpha > 2$.

D.3.3.2 $a_1 \geqslant 1$ or $a_2 \geqslant 1$

Now assume that at least one of $\{a_1, a_2\}$ is active. This means that we can replace the constraint in (D99) (after scaling) by

$$\frac{1}{\alpha}a_1 \;+\; \frac{1-y}{1-x}\frac{1}{\alpha}a_2 \;+\; \frac{y}{1-x}a_3 \;+\; \frac{x}{1-x}a_4 \;<\; \frac{1}{1-x} \tag{D102}$$

Substituting the extremal values of y and x gives the following constraints

$$\begin{cases} \frac{1}{\alpha}a_1 + \frac{\alpha+2}{(\alpha+1)(\alpha+1)}a_2 + \frac{\alpha+2}{(\alpha+1)(\alpha+1)}a_3 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1}, & (y = \frac{1}{\alpha+1}) \text{ and } (x = \frac{1}{\alpha+2}) \\ \frac{1}{\alpha}a_1 + \frac{1}{\alpha}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1}, & (y = \frac{1-x}{\alpha+1}) \text{ and } (x = \frac{1}{\alpha+2}) \\ \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_3 + \frac{1}{\alpha+2}a_4 < \frac{\alpha+3}{\alpha+2}, & (y = \frac{1}{\alpha+1}) \text{ and } (x = \frac{1}{\alpha+3}) \\ \frac{1}{\alpha}a_1 + \frac{\alpha^2+3\alpha+1}{\alpha(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 < \frac{\alpha+3}{\alpha+2}, & (y = \frac{1-x}{\alpha+1}) \text{ and } (x = \frac{1}{\alpha+3}) \end{cases} \tag{D103}$$

We now solve the resulting four programs.

$y = \frac{1}{\alpha+1}$ and $x = \frac{1}{\alpha+2}$

This gives the following program:

$$\mathcal{R}_1 = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \frac{1}{\alpha}a_1 + \frac{\alpha+2}{(\alpha+1)^2}a_2 + \frac{\alpha+2}{(\alpha+1)^2}a_3 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1} \\ & a_i \in \mathbb{N} \end{array}} = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \frac{1}{\alpha}a_1 + \frac{\alpha+2}{(\alpha+1)^2}a_2 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1} \\ & a_i \in \mathbb{N} \end{array}} \quad \text{(D104)}$$

We may assume that $a_3 = 0$, since it is dominated by a_2, giving the program on the RHS. If $a_1 \geqslant \alpha$ then additionally $a_2 = a_4 = 0$ and $\mathcal{R}_1 = 1$. Now add the constraint $a_1 \leqslant \alpha - 1$, and derive an upper bound for the resulting knapsack problem by the greedy heuristic. The relative profit of the items is given by $\rho(1) = 1$, $\rho(2) = \frac{(\alpha+1)(\alpha+3)}{(\alpha+2)^2}$ and $\rho(4) = \frac{\alpha+1}{\alpha+2}$. Note that $\rho(1) > \rho(2) > \rho(4)$. This gives

$$\mathcal{R}_1 < \frac{\alpha-1}{\alpha} + \frac{\alpha+3}{(\alpha+1)(\alpha+2)} + \rho(4)\left[\frac{\alpha+2}{\alpha+1} - \frac{\alpha-1}{\alpha} - \frac{\alpha+2}{(\alpha+1)^2}\right] = \cdots = 1 + \frac{\alpha-1}{\alpha(\alpha+1)} + \frac{1}{\alpha(\alpha+1)(\alpha+2)}$$

But since $\alpha(\alpha+1)(\alpha+2)$ is a scalar for the objective function, it follows that $\mathcal{R}_1 \leqslant 1 + \frac{\alpha-1}{\alpha(\alpha+1)}$ must hold. This bound is tight as shown by $\mathbf{a}^\mathsf{T} = [\alpha-1, 1, 0, 1]$. It is easy to show that $\mathcal{R}_1 \leqslant \mathcal{S}_\alpha(\alpha+2)$ holds.

$y = \frac{1-x}{\alpha+1}$ and $x = \frac{1}{\alpha+2}$

This gives the following program:

$$\mathcal{R}_2 = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \frac{1}{\alpha}a_1 + \frac{1}{\alpha}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+1}a_4 < \frac{\alpha+2}{\alpha+1} \\ & a_i \in \mathbb{N} \end{array}} = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{1}{\alpha+1}a_3 \\ \text{st} & \frac{1}{\alpha}a_1 + \frac{1}{\alpha+1}a_3 < \frac{\alpha+2}{\alpha+1} \\ & a_i \in \mathbb{N} \end{array}} \quad \text{(D105)}$$

We can eliminate variables a_2 and a_4, since they are dominated by a_1 and a_3 respectively. This gives a subset-sum problem with value $1 + \frac{\alpha-1}{\alpha(\alpha+1)}$, achieved for $\mathbf{a}^\mathsf{T} = [\alpha-1, 0, 2, 0]$. And, as in the previous case $\mathcal{R}_2 \leqslant \mathcal{S}_\alpha(\alpha+2)$ holds.

$x = \frac{1}{\alpha+3}$

We could explicitly formulate the programs $\mathcal{R}_3$ and $\mathcal{R}_4$, that correspond to the two subcases for y. However, it turns out that it is sufficient to prove an upper bound for their value. We do this by relaxing the respective constraints as follows.

- For the case $y = \frac{1}{\alpha+1}$ we reduce the multiplicand of a_3; $\frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_3 \rightarrow \frac{1}{\alpha+1}a_3$
- For the case $y = \frac{1-x}{\alpha+1}$ we reduce the multiplicand of a_2; $\frac{\alpha^2+3\alpha+1}{\alpha(\alpha+1)(\alpha+2)}a_2 \rightarrow \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2$

This gives the following subset-sum problem as an upper bound for both cases.

$$\mathcal{R}_5 = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 \\ \text{st} & \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+1}a_3 + \frac{1}{\alpha+2}a_4 < \frac{\alpha+3}{\alpha+2} \\ & a_i \in \mathbb{N} \end{array}} \tag{D106}$$

A scalar for the LHS of the constraint is given by $\alpha(\alpha+1)(\alpha+2)/\gcd\{\alpha,2\}$, which implies that we can tighten the RHS to $\frac{\alpha+3}{\alpha+2} - \frac{\gcd\{\alpha,2\}}{\alpha(\alpha+1)(\alpha+2)}$. Since it is a subset-sum problem, this is also a bound for its value. And thus $\mathcal{R}_3, \mathcal{R}_4 \leqslant \mathcal{S}_\alpha(\alpha+2)$ holds.

D.3.4 $\alpha \geqslant 3$, $\beta = \alpha + 2$ and there is no bin with largest item $y \in \left\langle \frac{1-x}{\alpha+1}, \frac{1}{\alpha+1} \right]$

The maximum pattern-weight, under weighting function (8.88), is given by the following program:

$$\mathcal{R} = \boxed{\begin{array}{ll} \text{Max} & \frac{1}{\alpha}a_1 + \frac{\alpha+1}{\alpha(\alpha+2)}a_2 + \frac{1}{\alpha+2}a_3 \\ \text{st} & \left(\frac{1-x}{\alpha}\right)^+ a_1 + \left(\frac{1}{\alpha+1}\right)^+ a_2 + x a_3 \leqslant 1 \\ & x \in \left\langle \frac{1}{\alpha+3}, \frac{1}{\alpha+2} \right] \text{ and } a_i \in \mathbb{N} \end{array}} \tag{D107}$$

A solution, feasible for all $x < \frac{1}{\alpha+1}$, is given by $\mathbf{a}^\mathsf{T} = [\alpha-1, 1, 1]$ with value $1 + \frac{\alpha-1}{\alpha(\alpha+2)}$. This is in fact the optimal solution, as shown by the following case analysis.

If $a_1 = a_2 = 0$ then $\mathcal{R} = 1$, so assume that at least one of $\{a_1, a_2\}$ is non-zero. This means that we can replace the constraint (after scaling) by $\frac{1}{\alpha}a_1 + \frac{1}{1-x}\frac{1}{\alpha+1}a_2 + \frac{x}{1-x}a_3 < \frac{1}{1-x}$. Substituting the extremal values of x gives the following constraints, and two cases to investigate.

$$\begin{cases} \frac{1}{\alpha}a_1 + \frac{\alpha+3}{(\alpha+1)(\alpha+2)}a_2 + \frac{1}{\alpha+2}a_3 < \frac{\alpha+3}{\alpha+2}, & (x = \frac{1}{\alpha+3}) \\ \frac{1}{\alpha}a_1 + \frac{\alpha+2}{(\alpha+1)^2}a_2 + \frac{1}{\alpha+1}a_3 < \frac{\alpha+2}{\alpha+1}, & (x = \frac{1}{\alpha+2}) \end{cases} \tag{D108}$$

$\boldsymbol{x = \frac{1}{\alpha+3}}$

In the corresponding program, the relative profit of the items is: $\rho(1) = 1$, $\rho(2) = \frac{(\alpha+1)^2}{\alpha(\alpha+3)}$ and $\rho(3) = 1$. Note that $1 = \rho(1) = \rho(3) > \rho(2)$. And thus $\mathcal{R} < \frac{\alpha+3}{\alpha+2}$, but since $\alpha(\alpha+2)$ is a scalar for the objective function, we can tighten this to $\mathcal{R} \leqslant \frac{\alpha+3}{\alpha+2} - \frac{1}{\alpha(\alpha+2)} = 1 + \frac{\alpha-1}{\alpha(\alpha+2)}$.

$\boldsymbol{x = \frac{1}{\alpha+2}}$

In the corresponding program, the relative profit of the items is: $\rho(1) = 1$, $\rho(2) = \frac{(\alpha+1)^3}{\alpha(\alpha+2)^2}$ and $\rho(3) = \frac{\alpha-1}{\alpha+2}$. Note that $1 = \rho(1) > \rho(2) > \rho(3)$. If $a_1 \geqslant \alpha$ then $a_2 = a_3 = 0$ and $\mathcal{R} = 1$. Now add the constraint $a_1 \leqslant \alpha - 1$, and derive an upper bound for the resulting knapsack problem by the greedy heuristic.

$$\mathcal{R} < \frac{\alpha-1}{\alpha} + \frac{\alpha+1}{\alpha(\alpha+2)} + \rho(3)\left[\frac{\alpha+2}{\alpha+1} - \frac{\alpha-1}{\alpha} - \frac{\alpha+2}{(\alpha+1)^2}\right] = \cdots = 1 + \frac{\alpha-1}{\alpha(\alpha+2)} + \frac{1}{\alpha(\alpha+1)(\alpha+2)}$$

But, since $\alpha(\alpha+2)$ is a scalar for the objective function, it follows that $\mathcal{R} \leqslant 1 + \frac{\alpha-1}{\alpha(\alpha+2)}$ must hold.

Appendix E

Miscellaneous

E.1 Function $S(n)$

In this section we will examine the function $S(n) = \sum_{i=2}^{n} \frac{\text{sp}(i)-1}{i}$, where $\text{sp}(i)$ is defined as the smallest prime factor of i. The results of this section are used in chapter 4, where the harmonic CSP is studied.

E1 **Property** $S(n) \leqslant \frac{n}{2} - \frac{1}{n}$

Proof. *By induction. The property clearly holds for* $n = 2$ *and* $n = 3$, *since* $S(2) = \frac{1}{2}$ *and* $S(3) = \frac{7}{6}$. *Now assume that the property holds for* n *and distinguish between the following two cases.*

n is odd: $S(n+1) = S(n) + \frac{1}{n+1} \leqslant \frac{n}{2} - \frac{1}{n} + \frac{1}{n+1} = \frac{n+1}{2} - \frac{1}{n+1} - \left[\frac{1}{2} + \frac{1}{n} - \frac{2}{n+1}\right] < \frac{n+1}{2} - \frac{1}{n+1}.$

n is even: $S(n+1) \leqslant S(n-1) + \frac{1}{n} + \frac{n}{n+1} \leqslant \frac{n-1}{2} - \frac{1}{n-1} + \frac{1}{n} + \frac{n}{n+1} = \frac{n+1}{2} - \frac{1}{n+1} - \frac{1}{n(n-1)} < \frac{n+1}{2} - \frac{1}{n+1}.$

So, if the property holds for n *it also holds for* $n+1$, *which completes the proof.* □

E2 **Property** $S(n) \leqslant 1 + \frac{n-1}{3}$ for $n \geqslant 32$

Proof. *By induction. To establish a recursion we use that any set of consecutive numbers* $\{n+1, \ldots, n+30\}$ *with* $n \in \mathbb{N}^+$ *contains exactly* 15 *numbers with smallest prime factor* 2, 5 *numbers with smallest prime factor* 3, 2 *numbers with smallest prime factor* 5 *and* 8 *numbers with smallest prime factor* $\geqslant 7$. *Now assume that the property holds for* n *and bound* $S(n+30)$ *as follows.*

$$\begin{aligned} S(n+30) &= \sum_{i=1}^{30} \frac{\text{sp}(n+i)-1}{n+i} + S(n) \leqslant 15 \times \frac{1}{n+1} + 5 \times \frac{2}{n+1} + 2 \times \frac{4}{n+1} + 8 \times (1 - \frac{1}{n+30}) + S(n) \\ &\leqslant \frac{33}{n+1} + 8 - \frac{8}{n+30} + 1 + \frac{n-1}{3} \leqslant 1 + \frac{(n+30)-1}{3}, \quad \text{for } n \geqslant 15 \end{aligned}$$

So, if the property holds for a particular $n \geqslant 15$ *it also holds for* $n+30$. *Direct comparison shows that the property holds for* $n \in \{32, \ldots, 61\}$ *which, combined with the recursion, proves the claim. We note that the only values of* n *for which the property does not hold are* 7, 8, 11–15, 17–25 *and* 31. □

To establish the asymptotic behaviour of $S(n)$ and highlight the slow convergence of $S(n)/n$ we derive the following properties, where $\pi(n)$ is the prime counting function.

E3 Property $-\frac{5}{6} \leqslant S(n) - \pi(n) \leqslant \sqrt{n}$

Proof. *Denote by $r(n)$ the remainder term $S(n) - \pi(n)$, so that $r(n) = \sum_{c \leqslant n} \frac{\mathrm{sp}(i)}{i} - \sum_{i=2}^{n} \frac{1}{i}$, where the summation is taken over all composite numbers less than or equal to n.*

We first prove the upper bound $r(n) \leqslant \sqrt{n}$ by induction. It is easily verified that for $n \leqslant 5$ this bound holds. Now distinguish between the following cases.

n is even: $r(n+2) \leqslant r(n) + \frac{1}{\sqrt{n+1}} + \frac{2}{n+2} - \frac{1}{n+1} - \frac{1}{n+2} < \sqrt{n} + \frac{1}{\sqrt{n+1}}$

n is odd: $r(n+2) \leqslant r(n) + \frac{2}{n+1} + \frac{1}{\sqrt{n+2}} - \frac{1}{n+1} - \frac{1}{n+2} < \sqrt{n} + \frac{1}{\sqrt{n+2}} + \frac{1}{(n+1)(n+2)}$

For both cases the last expression in n is strictly less than $\sqrt{n+2}$ for $n \geqslant 2$, which completes the proof for the upper bound. We now prove the lower bound $r(n) \geqslant -\frac{5}{6}$. For $n \leqslant 16$ this is easily verified, with equality holding only for $n = 3$ (note that we need only check for n prime). For $n \geqslant 17$ the lower bound follows from

$$\begin{aligned} r(n) &= \sum_{c \leqslant n} \frac{\mathrm{sp}(i)}{i} - \sum_{i=2}^{n} \frac{1}{i} \geqslant \sum_{i=2}^{\lfloor n/2 \rfloor} \frac{1}{i} - \sum_{i=2}^{n} \frac{1}{i} = \sum_{i=8}^{\lfloor n/2 \rfloor} \frac{1}{i} - \sum_{i=8}^{n} \frac{1}{i} \\ &> \Big(\ln(\lfloor n/2 \rfloor + 1) - \ln 8\Big) - \Big(\ln n - \ln 7\Big) > -4\ln 2 + \ln 7 > -\tfrac{5}{6}. \end{aligned}$$

The first summation (over the composite numbers) is restricted to the even numbers in $\{4, \ldots, n\}$. The change from $i = 2$ to $i = 8$ is valid since $n \geqslant 17$. For the last part we have used the inequalities $\ln(b+1) - \ln a < \sum_{i=a}^{b} 1/i < \ln b - \ln(a-1)$ and $\lfloor n/2 \rfloor + 1 > n/2$. This proves the lower bound and thus the property. □

E4 Property $S(n) \sim \frac{n}{\ln n}$

Proof. *From the prime number theorem [23, Volume 3; 'Distribution of Prime Numbers'] we know that $\pi(n) \sim \frac{n}{\ln n}$, which combined with property E3 proves the claim.* □

The asymptotic behaviour of $S(n)$ and the slow convergence of $S(n)/\frac{n}{\ln n}$ can be verified in table E.1.

Comment

An early version of the proof of lemma 4.9 was based upon the following bound for the prime counting function due to Tschebycheff;

$$A\frac{n}{\ln n} < \pi(n) < B\frac{n}{\ln n}, \quad \text{for } n \geqslant n_0 \tag{E1}$$

where $A = \ln \frac{2^{1/2}3^{1/3}5^{1/5}}{30^{1/30}} \approx 0.92129$, $B = \frac{6}{5}A \approx 1.10555$ and $n_0 = 30$. This value for n_0 is given in [57, chapter 4; 'The growth of $\pi(x)$'] and [22, chapter 1; 'Tschebycheff's effective theorems']. However, this value for n_0 is incorrect. The only thing that can be said is that there exists an n_0 such that (E1) holds for all $n \geqslant n_0$. Other authors[23, 51, 58] refrain from giving explicit values for n_0. That n_0 cannot be equal to 30 follows from $\pi(30) = 10$ with 'upper bound' $B\frac{n}{\ln n} \approx 9.7514$. Moreover, from [58, table 3] it follows that n_0 must be larger than $80K$ since $\pi(80K) = 7837$ and $B\frac{n}{\ln n} \approx 7833.9896$.

n	$S(n)$	$1+\frac{S(n)-1}{n-1}$	$S(n)/\frac{n}{\ln n}$	n	$S(n)$	$1+\frac{S(n)-1}{n-1}$	$S(n)/\frac{n}{\ln n}$
–	–	–	–	100	26.643	1.259	1.227
2	0.500	0.500	0.173	200	48.637	1.239	1.288
3	1.167	1.083	0.427	300	65.415	1.215	1.244
4	1.417	1.139	0.491	400	81.995	1.203	1.228
5	2.217	1.304	0.714	500	99.443	1.197	1.236
6	2.383	1.277	0.712	600	113.881	1.188	1.214
7	3.240	1.373	0.901	700	130.205	1.185	1.219
8	3.365	1.338	0.875	800	144.547	1.180	1.208
9	3.588	1.324	0.876	900	159.853	1.177	1.208
10	3.688	1.299	0.849	1000	174.122	1.173	1.203
20	7.865	1.361	1.178	2000	311.164	1.155	1.183
30	10.311	1.321	1.169	3000	439.546	1.146	1.173
40	12.617	1.298	1.164	4000	560.647	1.140	1.163
50	15.824	1.303	1.238	5000	680.558	1.136	1.159
60	18.025	1.289	1.230	6000	795.373	1.132	1.153
70	20.192	1.278	1.226	7000	913.054	1.130	1.155
80	23.322	1.283	1.277	8000	1020.715	1.127	1.147
90	25.452	1.275	1.273	9000	1131.282	1.126	1.144
–	–	–	–	10000	1243.809	1.124	1.146

Table E.1. Function $S(n)$ and its asymptotic behaviour. The quantity $1+\frac{S(n)-1}{n-1}$ provides an upper bound for the gap of a harmonic CSP (see section 4.3) and converges to 1. The quantity $S(n)/\frac{n}{\ln n}$ also converges to 1.

E.2 Harmonic bound for First-Fit Decrease.

In this section we prove a bound for FFD, which is based on the NFD-weighting function (7.6). This bound will be used in chapter 4, where the harmonic CSP is studied. Denote by n_i the number of items in the interval $\langle\frac{1}{i+1},\frac{1}{i}]$. Note that n_i can be zero.

E5 **Lemma** $FFD < 2+\sum n_i/i$

Proof. *Suppose that we have a list $\mathcal{L}$ with $f_i \geqslant 1$ items in the intervals $\langle\frac{1}{\alpha_i+1},\frac{1}{\alpha_i}]$, where $1 \leqslant i \leqslant m$ and $1 \leqslant \alpha_1 < \cdots < \alpha_m$, so that $\alpha_m \geqslant m$. We derive a bound for $c(\mathcal{L}) = FFD - W(\mathcal{L})$ under the weighting function $W(s) = 1/\lfloor 1/s\rfloor$. Wlog we may assume that the last bin is a singleton bin with an item $x \in \langle\frac{1}{\alpha_m+1},\frac{1}{\alpha_m}]$. All other bins are therefore filled to a level $> 1-x$. This implies, since $W(s) \geqslant s$, that the bin weight W_j of those bins is $W_j > 1-x \geqslant 1-1/\alpha_m$.*

Now define δ_i as the number of bins with an α_i-item as largest item and bin weight strictly less than 1. Suppose that there are at least two such bins. Then by the FFD-heuristic the first of these bins must have been filled to a level $> 1-1/\alpha_i$ before the first item in the second of those bins was placed. But this implies that the first of these bins contains exactly α_i α_i-items. This in turn means that it has a bin weight of at least 1. Contradiction, so $\delta_i = 0, 1$.

This means that for every $i \in \{1,\ldots,m\}$ there is at most one bin with bin weight strictly less than 1. We can now bound $c(\mathcal{L})$ as follows, where the term in square brackets is the weight of the singleton bin.

$$c(\mathcal{L}) = \sum_j (1 - W_j) \leqslant \sum_{j|W_j<1} (1 - W_j) \leqslant \sum_{i=1}^{m-1} 1/\alpha_m + [1 - 1/\alpha_m] \tag{E2}$$

An upper bound for $c(\mathcal{L})$ is thus $1 + \frac{m-2}{\alpha_m}$. Using $\alpha_m \geqslant m$ it follows that $c(\mathcal{L}) < 2$ must hold. This proves the bound FFD $< 2 + \sum f_i/\alpha_i$ and since $\sum f_i/\alpha_i = \sum n_i/i$ the lemma follows. □

That this bound is the best possible follows from the following lemma.

E6 Lemma *There are lists such that FFD $\geqslant 2 - \varepsilon + \sum n_i/i$, for all $\varepsilon > 0$.*

Proof. *By construction. We shall prove that for every integer $\alpha \geqslant 2$ there is a list such that*

$$2 - \tfrac{2}{\alpha} < FFD - \sum \tfrac{n_i}{i} < 2 - \tfrac{1}{\alpha} \tag{E3}$$

It is easy to see that (E3) proves the claim by choosing α such $\alpha \geqslant 2/\varepsilon$. Assume that we have chosen such an α and define $\beta = \alpha^2 + \alpha - 1$. Further define the integers $\{a_k\}$ by

$$\tfrac{k(k+1)}{\beta} - 1 \leqslant a_k < \tfrac{k(k+1)}{\beta}, \qquad \alpha \leqslant k \leqslant \beta - 1. \tag{E4}$$

$$a_{\alpha-1} \equiv 0 \text{ and } a_\beta \equiv \beta - 1 \tag{E5}$$

With this we can define $n_k = a_{k-1} + k - a_k$ for $k \in \{\alpha,\ldots,\beta\}$. The list that proves (E3) consists of n_k items of size $1/k$. We claim that FFD uses $\beta - \alpha + 1 = \alpha^2$ bins to pack this list and that these bins are packed as follows. To facilitate the notation we label these bins as $\alpha,\ldots,\beta$. Bin $k \in \{\alpha,\ldots,\beta-1\}$ is packed with $(k - a_k)$ items of size $1/k$ and a_k items of size $1/(k+1)$. Bin β is packed with a singleton item of size $1/\beta$. We prove the claim by induction.

(i) That the claim holds for bin α is easily verified. We have $a_\alpha = 1$, so that $n_\alpha = \alpha - 1$. FFD will pack all of the $\alpha - 1$ items of size $1/\alpha$ and 1 item of size $1/(\alpha+1)$ in this bin.

(ii) Suppose that the claim holds for all bins with an index lower than k. Then, after packing the bins α to $k-1$, there are $(k - a_k)$ items of size $1/k$ and n_j items of size $1/j$, where $k+1 \leqslant j \leqslant \beta$, left to pack. Since

$$l_k = (k - a_k) \times \tfrac{1}{k} + a_k \times \tfrac{1}{k+1} = 1 - \tfrac{a_k}{k(k+1)} > 1 - \tfrac{1}{\beta}$$

FFD will pack exactly $(k - a_k)$ items of size $1/k$ and a_k items of size $1/(k+1)$ in bin k.

(iii) After packing bin $(\beta - 1)$ there are exactly $\beta - a_\beta = 1$ items of size $1/\beta$ left. This item is packed in bin β as a singleton.

Every number in $\{\alpha,\ldots,\beta\}$ corresponds to exactly one bin packed, so that FFD uses $\beta - \alpha + 1 = \alpha^2$ bins. This completes the proof of the claim.

We will now derive an expression for $c(\mathcal{L}) = FFD - \sum n_k/k$.

$$\begin{aligned} c(\mathcal{L}) &= FFD - \sum \tfrac{n_k}{k} = (\beta-\alpha+1) - \sum_{k=\alpha}^{\beta} \left[\tfrac{a_{k-1}}{k} + 1 - \tfrac{a_k}{k}\right] = \sum_{k=\alpha}^{\beta} \tfrac{a_k}{k} - \sum_{k=\alpha}^{\beta} \tfrac{a_{k-1}}{k} \\ &= \sum_{k=\alpha}^{\beta} \tfrac{a_k}{k} - \sum_{k=\alpha-1}^{\beta-1} \tfrac{a_{k-1}}{k+1} = \tfrac{a_\beta}{\beta} + \sum_{k=\alpha}^{\beta-1} \tfrac{a_k}{k(k+1)} \end{aligned} \tag{E6}$$

From which we get the following bounds.

$$\text{Upper bound:} \quad c(\mathcal{L}) < 1 - 1/\beta + \sum_{k=\alpha}^{\beta-1} \tfrac{1}{\beta} = 2 - \tfrac{\alpha+1}{\beta} < 2 - \tfrac{1}{\alpha} \tag{E7}$$

$$\begin{aligned} \text{Lower bound:} \quad c(\mathcal{L}) &\geqslant 1 - 1/\beta + \sum_{k=\alpha}^{\beta-1} \left[\tfrac{1}{\beta} - \tfrac{1}{k(k+1)}\right] = 2 - \tfrac{\alpha+1}{\beta} - \sum_{k=\alpha}^{\beta-1} \tfrac{1}{k(k+1)} \\ &= 2 - \tfrac{\alpha+1}{\beta} - \sum_{k=\alpha}^{\beta-1} \tfrac{1}{k} + \sum_{k=\alpha}^{\beta-1} \tfrac{1}{k+1} = 2 - \tfrac{\alpha+1}{\beta} - \tfrac{1}{\alpha} + \tfrac{1}{\beta} = 2 - \tfrac{\alpha}{\beta} - \tfrac{1}{\alpha} \\ &> 2 - \tfrac{2}{\alpha} \end{aligned} \tag{E8}$$

These bounds prove (E3) and thus the lemma. □

We finish by giving some examples (table E.2) of the lists that were constructed to satisfy (E3). Packing one of these lists by the FFD-algorithm, as is done in diagram E.1, will serve to illustrate the structure of the packing and the rationale behind the proof.

Comments

- Note that the lists, that show that the bound in lemma E5 is the best possible, are harmonic. However, since FFD is not monotonic, we cannot make the a priori assumption that a list that maximises $FFD(\mathcal{L}) - \sum n_i/i$ consists of items of size $1/i$.
- It was later discovered that Terno & Scheithauer[61] have given a proof for $OPT \leqslant 1 + \lceil \sum n_i/i \rceil$. Their proof is based on an induction argument and is not as straightforward as the proof of lemma E5.
- lemma E5 provides us with an easily calculable, and more problem-specific performance bound for FFD; $FFD < 2 + rCSP_R$. Where $r = \max\{\sum a_i/\lfloor L/d_i \rfloor \mid \sum a_i d_i \leqslant L, \mathbf{a} \in \mathbb{N}^m\}$ is the NFD worst-case ratio for lists drawn from this particular set of items.

α	$\mathcal{L}$	$c(\mathcal{L})$
2	$\{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}\}$	$1\frac{17}{60}$ $\approx 1.28333...$
3	$\{\frac{2}{3}, \frac{4}{4}, \frac{4}{5}, \frac{5}{6}, \frac{5}{7}, \frac{7}{8}, \frac{7}{9}, \frac{9}{10}, \frac{10}{11}\}$	$1\frac{14521}{27720}$ $\approx 1.52384...$
4	$\{\frac{3}{4}, \frac{5}{5}, \frac{5}{6}, \frac{7}{7}, \frac{7}{8}, \frac{8}{9}, \frac{9}{10}, \frac{10}{11}, \frac{10}{12}, \frac{12}{13}, \frac{12}{14}, \frac{14}{15}, \frac{14}{16}, \frac{15}{17}, \frac{17}{18}, \frac{18}{19}\}$	$1\frac{683567}{1939938}$ $\approx 1.64763...$
5	$\{\frac{4}{5}, \ldots, \frac{28}{29}\}$	$\approx 1.72586...$
8	$\{\frac{7}{8}, \ldots, \frac{70}{71}\}$	$\approx 1.82041...$
14	$\{\frac{13}{14}, \ldots, \frac{208}{209}\}$	$\approx 1.90087...$
100	$\{\frac{99}{100}, \ldots, \frac{10098}{10099}\}$	$\approx 1.98549...$

Table E.2. Examples to show the convergence of $c(\mathcal{L}) = FFD(\mathcal{L}) - \sum n_i/i$ for lists generated by (E4) and (E5). Note that i/j denotes i items of size $1/j$.

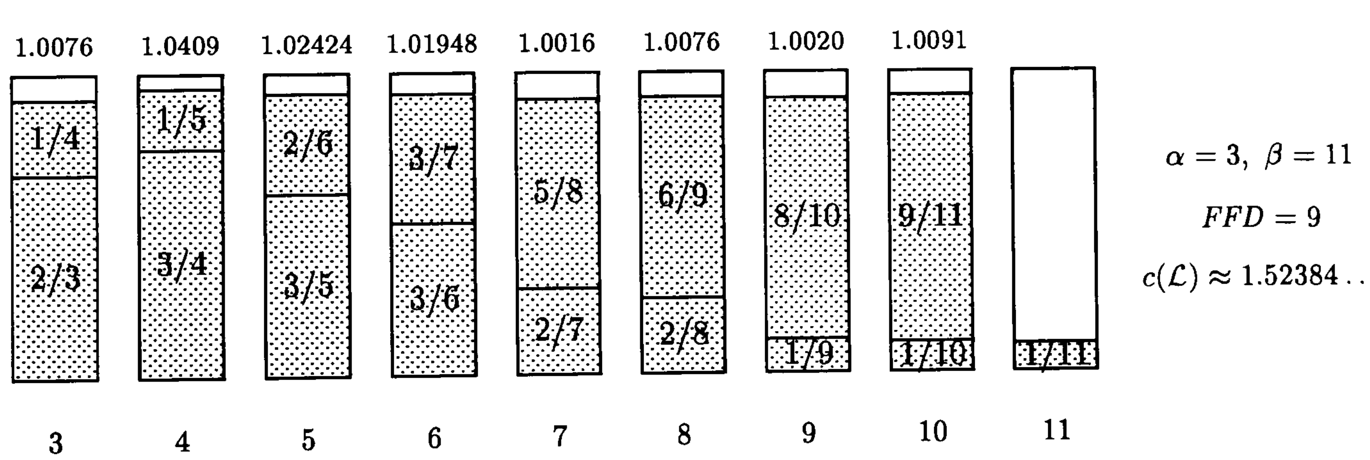


Diagram E.1. Example to show the structure of packings of harmonic lists by FFD, used in the proof of lemma E6. Bin $k \in \{\alpha, \ldots, \beta - 1\}$ is filled exclusively with items of size $1/k$ and $1/(k+1)$. The number of each is chosen as to minimise the bin-level l_k, subject to $l_k + 1/\beta > 1$. This implies that bin k contains a total of k items. The number on top of a bin indicates to what level the bin would be filled if the singleton item would be inserted. Note that (in the diagram) i/j denotes i items of size $1/j$.

E.3 Harmonic bound for Next-Fit Decrease.

In this section we prove a bound for *NFD* in terms of the number of i-items. The objective is not so much the bound itself, but corollary E8, which will be used in chapter 7 to construct examples to show that the asymptotic ratios are achievable. Denote by f_i the number of items in the interval $\langle \frac{1}{i+1}, \frac{1}{i}]$. Note that f_i can be zero.

E7 **Lemma** *If f_i is the number of i-items then* $\sum_i \left\lfloor \frac{f_i}{i} \right\rfloor \leqslant NFD \leqslant \sum_i \left\lceil \frac{f_i}{i} \right\rceil$

Proof. *Define n_i as the number of i-bins. The f_i i-items will be packed into $\delta_1 + k + \delta_2$ consecutive bins as shown in diagram E.2, where δ_1 is defined as 1 if the first bin with an i-item is not i-complete and 0*

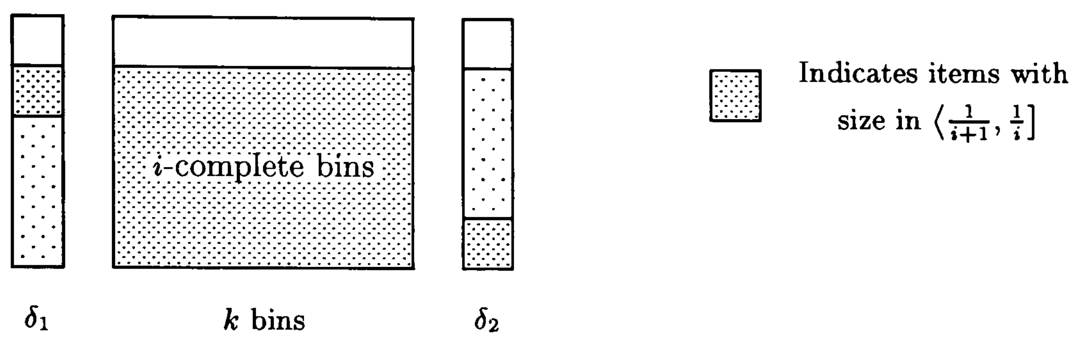


Diagram E.2. *NFD-packing of i-items.*

otherwise. Similarly, δ_2 is defined as 1 if the last bin with an i-item is not i-complete and 0 otherwise. Further, define r_1 as the number of i-items in bin δ_1 and similarly r_2 as the number of i-items in bin δ_2. This gives $1 \leqslant r_1 \leqslant i-1$ and $1 \leqslant r_2 \leqslant i-1$. From the balance $f_i = r_1\delta_1 + ki + r_2\delta_2$ we derive

$$n_i = k + \delta_2 = \frac{f_i}{i} - \frac{r_1}{i}\delta_1 + \left(1 - \frac{r_2}{i}\right)\delta_2 \tag{E9}$$

This expression is maximised for $\delta_1 = 0, \delta_2 = 1$ and $r_2 = 1$, and minimised for $\delta_1 = 1, \delta_2 = 0$ and $r_1 = i-1$. Since n_i is integer we have

$$\left\lceil \frac{f_i - i + 1}{i} \right\rceil \leqslant n_i \leqslant \left\lfloor \frac{f_i + i - 1}{i} \right\rfloor \quad \Longrightarrow \quad \left\lfloor \frac{f_i}{i} \right\rfloor \leqslant n_i \leqslant \left\lceil \frac{f_i}{i} \right\rceil \tag{E10}$$

and the lemma follows from $NFD = \sum_i n_i$. □

Note that lemma E7 implies that $NFD \sim \sum f_i/i$, or more specifically that $|NFD - \sum f_i/i| < m$. The special case when all f_i are multiples of i gives the following corollary.

E8 **Corollary** *If f_i is the number of i-items and $\forall i \; f_i \equiv 0 \pmod{i}$ then $NFD = \sum f_i/i$.*

Note: The rationale of the proof of lemma E7 can be applied to NFD-heuristics for higher dimensional problems. If the heuristic packs m_i items in an i-complete bin then $1 \leqslant r_{1,2} \leqslant m_i - 1$ hold. This applied to lemma E7 gives $\sum\lfloor f_i/m_i \rfloor \leqslant H \leqslant \sum\lceil f_i/m_i \rceil$, and a corollary similar to corollary E8; viz. $\forall i \; f_i \equiv 0 \pmod{m_i} \Rightarrow H = \sum f_i/m_i$. Example: the Bottom-Left Next-Fit Decreasing heuristic, when used to pack squares has $m_i = i^2$ and thus yields $\sum\lfloor f_i/i^2 \rfloor \leqslant BLNFD \leqslant \sum\lceil f_i/i^2 \rceil$.

E.4 Bounds for Next-Fit.

The *Next-Fit (NF)* heuristic takes a list $\mathcal{L}$ and places each item, in succession, in the current bin. When an item cannot be placed, the current bin is closed and a new bin is opened, which then becomes the current bin, and in which this item is placed. As an illustration, see the examples in diagrams E.3 and E.4.

If m items with sizes x_i are packed in a bin, we may assume wlog that they are in decreasing order; that is $x_1 \geqslant \cdots \geqslant x_m$. This leads to the following characteristics of a NF-packing, where w_j is the wastage, and l_j is the largest item in bin j.

- **Invariant:** $w_{j-1} < l_j,\ 2 \leqslant j \leqslant NF$
- **Recurrent bin:** $w_j < l_j \Rightarrow \sum_{i=1}^{m} x_i > 1 - x_1$

We can use the recurrency as a guide to find suitable weighting functions. Denote by φ the size of the largest item in the list. For $\varphi > \frac{1}{2}$ we can use $W(s) = 2s$ as a weighting function, which is easily seen to be recurrent. For $\varphi \leqslant \frac{1}{2}$ we can use the weighting function $W(s) = \frac{1}{1-\varphi}s$. Since all recurrent bins have $\sum x_i > 1 - x_1 \geqslant 1 - \varphi$, it follows that this function is recurrent. We can now derive bounds in the following lemmas by considering $c(\mathcal{L}) = NF(\mathcal{L}) - W(\mathcal{L})$ as a maximisation problem over all feasible lists.

E9 Lemma $\varphi > \frac{1}{2} \ \Rightarrow\ NF < 2Mat + \begin{cases} -1 + 2\varphi, & \textit{if NF is odd} \\ 0, & \textit{if NF is even} \end{cases}$

Proof. *We consider $c(\mathcal{L}) = NF(\mathcal{L}) - 2Mat(\mathcal{L})$. First we will prove the lemma for $NF(\mathcal{L}) \leqslant 3$.*

(1) $NF(\mathcal{L}) = 1 \Rightarrow Mat \geqslant \varphi \Rightarrow c \leqslant 1 - 2\varphi$

(2) $NF(\mathcal{L}) = 2 \Rightarrow Mat > 1 \Rightarrow c < 0$

(3) $NF(\mathcal{L}) = 3 \Rightarrow Mat \geqslant (1 - w_1) + (1 - w_2) + l_3 > 2 - w_1 > 2 - l_2 \geqslant 2 - \varphi \Rightarrow c < -1 + 2\varphi.$

For the first case, note that $1 - 2\varphi < -1 + 2\varphi$ holds for $\varphi > \frac{1}{2}$. This proves the lemma for $NF(\mathcal{L}) \leqslant 3$. Now assume that $NF(\mathcal{L}) \geqslant 4$ and make the following observation. The material packed in the first two bins is at least $(1 - w_1) + l_2 > 1$ (use the invariant). This implies that we can create a smaller list with larger c-value, by deleting the [items, packed in the] first two bins. Repeating this means that, if NF is odd we end up with a list with $NF(\mathcal{L}) = 3$, and if NF is even we end up with $NF(\mathcal{L}) = 2$. For both these cases the lemma holds, which completes the proof. □

E10 Lemma $\varphi \leqslant \frac{1}{2}$ and $Mat > \varphi \ \Rightarrow\ NF < 2 - \frac{1}{1-\varphi} + \frac{1}{1-\varphi}Mat.$

Proof. *We consider $c(\mathcal{L}) = NF(\mathcal{L}) - \frac{1}{1-\varphi}Mat(\mathcal{L})$. First we will prove the lemma for $NF(\mathcal{L}) \leqslant 2$.*

(1) $NF(\mathcal{L}) = 1 \Rightarrow Mat > \varphi \Rightarrow c < 1 - \frac{\varphi}{1-\varphi}$

(2) $NF(\mathcal{L}) = 2 \Rightarrow Mat > 1 \Rightarrow c < 2 - \frac{1}{1-\varphi}$

This proves the lemma for $NF(\mathcal{L}) \leqslant 2$. Now assume that $NF(\mathcal{L}) \geqslant 3$ and make the following observation. The material packed in the first bin is at least $1 - w_1 > 1 - l_2 \geqslant 1 - \varphi$. This implies that we can create a smaller list with larger c-value, by deleting the [items, packed in the] first bin. Repeating this means that we end up with a list with $NF(\mathcal{L}) = 2$. For this case the lemma holds, which completes the proof. □

We can now use the lemmas to derive the following corollaries. Substitution of $\varphi = 1$ in lemma E9 gives $\lfloor NF/2 \rfloor < Mat$, and leads to the first corollary. The second one is basically a reformulation of lemma E10. The third one follows from the previous two and the fact that for $Mat \leqslant 1$ we have $NF = OPT = 1$ by (3.1).

E11 Corollary $\varphi > \frac{1}{2} \;\Rightarrow\; NF \leqslant -1 + 2\lceil Mat \rceil$

E12 Corollary $\varphi \leqslant \frac{1}{2}$ and $Mat > 1 \Rightarrow\; NF \leqslant 1 + \lceil \frac{Mat-1}{1-\varphi} \rceil \leqslant -1 + 2\lceil Mat \rceil$

E13 Corollary $NF \leqslant -1 + 2OPT$

Bounds are tight

That the bounds in the first two lemmas are the best possible follows from the instances shown in diagrams E.3 and E.4. Choosing ε sufficiently small, or more specifically $\varepsilon = o(1/k)$ shows that we can approximate the bounds as closely as desired. There is one 'caveat'; we need to choose ε such that $1/\varepsilon$ and $(1 - \varphi)/\varepsilon$ are integer. Both requirements are taken care of by choosing $\varepsilon = \gcd(1, \varphi)/k^2$.

The first list also shows that the bound in lemma E11 is tight. It is easily verified that, for $\varepsilon = o(1/k)$, the list can be packed into $k + 1$ bins.

Previous bounds

Johnson[38] derived the following bounds.

$$-2 + 2OPT \leqslant \quad NF \quad \leqslant 1 + 2OPT \qquad \text{(E11)}$$

$$-\tfrac{1}{1-\varphi} + \tfrac{1}{1-\varphi}OPT < \quad NF \quad < 1 + \tfrac{1}{1-\varphi}OPT \qquad \text{for } \varphi \leqslant \tfrac{1}{2} \qquad \text{(E12)}$$

The lower bounds are to be interpreted as 'there is a list such that' and the upper bounds are to be interpreted as 'for all lists'. The first upper bound was later improved by Coffman et al.[12] to $NF \leqslant 2OPT$. This is the bound that is usually quoted in the literature.[28,49] More recently, Coffman et al.[14] give the tight bound $NF \leqslant -1 + 2OPT$, although without [reference to] a proof. Such a proof can be found in [36, p. 1551]

Comment

Note that a minimal list for NF packs into a configuration with only two different item-sizes per bin. This is because we can 'cut' any, but the first-placed, item in a bin into unit-sizes ε (see for example diagram E.4).

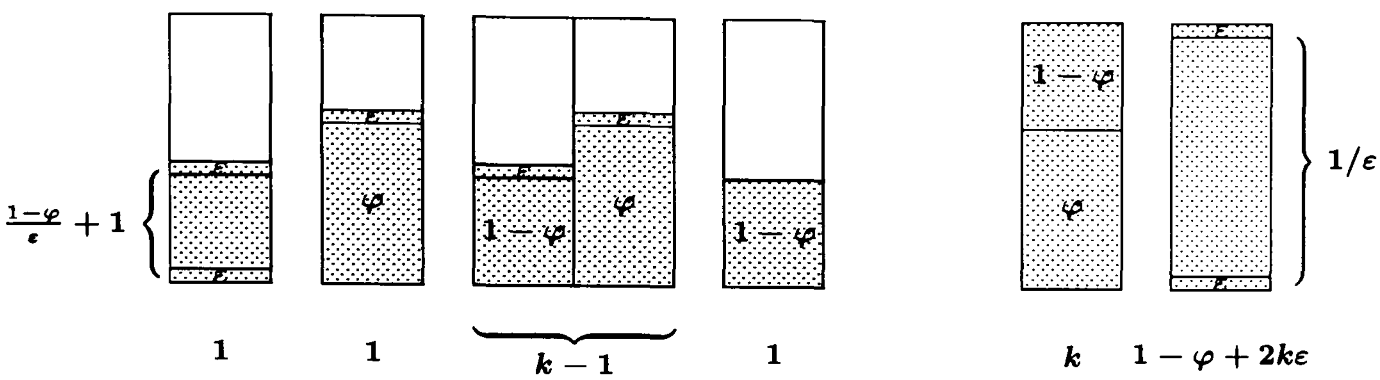


Diagram E.3. Worst-case example for NF for largest item $\varphi \geqslant \frac{1}{2}$.
$NF = 2k+1$ and $CSP_R = k+1-\varphi+2k\varepsilon$, which gives a constant $c(\mathcal{L}) = -1+2\varphi-4\varepsilon k$.
Note that ε is chosen such that $(1-\varphi)/\varepsilon$ and $1/\varepsilon$ are integer.

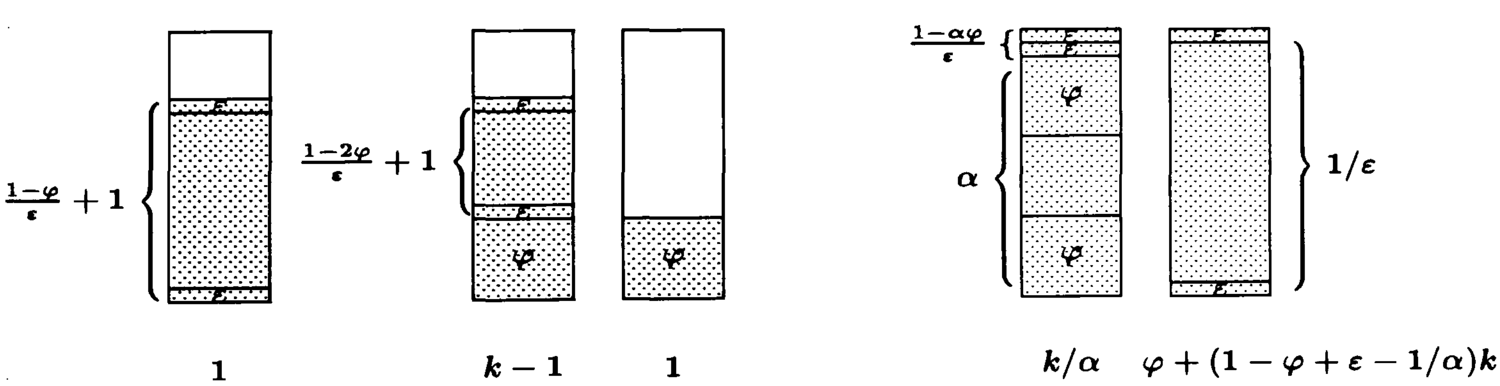


Diagram E.4. Worst-case example for NF for largest item $\varphi \leqslant \frac{1}{2}$.
$NF = k+1$ and $CSP_R = \varphi+(1-\varphi+\varepsilon)k$, which gives $c(\mathcal{L}) = 2 - \frac{1}{1-\varphi} - \frac{\varepsilon k}{1-\varphi}$.
Note that $\alpha = \lfloor 1/\varphi \rfloor$ and that ε is chosen such that $(1-\varphi)/\varepsilon$ and $1/\varepsilon$ are integer.

E.5 Monotonicity

We follow the definition of Murgolo,[50] and say that a bin-packing heuristic is *monotonic* if deleting items and/or decreasing the size, can never increase the number of bins used by the heuristic. We say that $\mathcal{L}$ *dominates* $\mathcal{L}'$ and denote this by $\mathcal{L}' \leqslant_d \mathcal{L}$, to indicate that the list $\mathcal{L}'$ can be constructed from $\mathcal{L}$ by deleting or decreasing [the size of] items. This allows us to call any function f, defined on a list of items, monotonic if $\mathcal{L}' \leqslant_d \mathcal{L}$ implies $f(\mathcal{L}') \leqslant f(\mathcal{L})$. It is easy to see that the functions/algorithms Mat, CSP and OPT are monotonic.

To prove that NF is monotonic we take a slightly different view of the NF-algorithm. Say we have a sequence of n numbers, which we have to divide in (a minimal number of) N sections, such that the sum within each section does not exceed 1. It is easy to see that deleting an item anywhere in this sequence

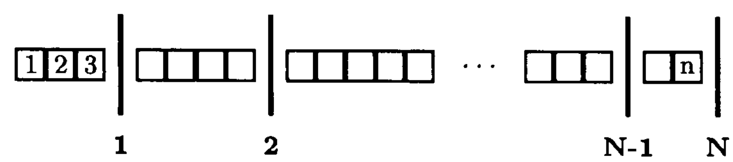


Diagram E.5. NF as a partitioning problem.

'pushes' the following items to the left, which means that the last section divider may become obsolete. This means that we need at most the same number of section dividers for the reduced list. The same holds when we decrease the size of an item, and conclude that NF is monotonic. A proof that NF is monotonic can also be found in Murgolo [50, Theorem 4.1.].

For NFD we have the added restriction that NFD keeps the numbers in decreasing sequence, so that changing the size of an item may alter the sequence. To prove the monotonicity of NFD first observe that any item n with size $x_n > x_{n+1}$ can be decreased to a size greater than or equal to x_{n+1} without altering the NFD-sequence. Now suppose that we want to decrease x_n by Δ and $\Delta > x_n - x_{n+1}$. This is the same as setting $x_n := x_{n+1}$ and decreasing x_{n+1} by $\Delta - (x_n - x_{n+1})$. This does not alter the sequence. We can repeat this process until we have reached either the end of the sequence or the remainder of Δ cannot be 'carried over'. For instance, decreasing an item of size 8 by 5 in the list $\{9, 8, 6, 4\}$ by this process produces $\{9, 8, 6, 4\} \to \{9, 6, 6, 4\} \to \{9, 6, 4, 4\} \to \{9, 6, 4, 3\}$. That *NFD* decreases when we delete an item of the list is easy to see, so that we can conclude that NFD is monotonic.

To prove that FF is not monotonic consider the following lists, based on the list in diagram 8.25 (p. 112).

$$\text{List } \mathcal{L} = \{ \underbrace{23, \ldots, 23}_{12k}, \underbrace{12, \ldots, 12}_{12k}, \underbrace{9, \ldots, 9}_{12k} \} \text{ on bins of size 44.} \tag{E13}$$

$$\text{List } \mathcal{L}' = \{ \underbrace{23, \ldots, 23}_{6k}, \underbrace{13, \ldots, 13}_{6k}, \underbrace{12, \ldots, 12}_{12k}, \underbrace{9, \ldots, 9}_{12k} \} \text{ on bins of size 44.} \tag{E14}$$

Take the list $\mathcal{L}$, which FF will pack into exactly $12k$ bins. Now decrease $6k$ items of size 23 to a size of 13 to get the list $\mathcal{L}'$. This list will be packed by FF into exactly $13k$ bins. From this we can conclude that FF is not monotonic. Moreover it demonstrates that $FF(\mathcal{L}') - FF(\mathcal{L})$ cannot be bound by a constant. Since the list $\mathcal{L}$ is already in decreasing order the same follows for FFD. Ergo neither FF nor FFD is monotonic.[1]

Extension

As an extension to the concept of [weak] monotonicity we allow one more operator to construct a dominated list. We say that a list $\mathcal{L}'$ is ***strongly dominated*** by a list $\mathcal{L}$, if $\mathcal{L}'$ can be constructed from $\mathcal{L}$ by deleting items and/or reducing their size and/or cutting an item into [and replacing it by these] smaller items, and denote this by $\mathcal{L}' \leqslant_{\mathrm{D}} \mathcal{L}$. To illustrate this, consider the lists $\mathcal{L} = \{\frac{1}{2}, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$, $\mathcal{L}' = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$ and $\mathcal{L}'' = \{\frac{1}{2}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{4}\}$. Then $\mathcal{L}$ strongly dominates $\mathcal{L}'$, but $\mathcal{L}$ does not strongly dominate $\mathcal{L}''$. This distinction is important for on-line heuristics. For off-line heuristics which sort the list before processing, it is only the set of items in the list that is important and not the sequence in which they appear. In analogy to a monotonic function, we say that a function f, defined on a list of items, is ***strongly monotonic*** if $\mathcal{L}' \leqslant_{\mathrm{D}} \mathcal{L}$ implies $f(\mathcal{L}') \leqslant_{\mathrm{D}} f(\mathcal{L})$.

That NF is strongly monotonic can be seen as follows. If we cut any item, which is not the first item in the bin, the packing does not change. If we cut the first item in say bin n, then some parts [but not all] of it may be packed in the previous bin. The packing of bins $n, n+1, \ldots$ now is the same as that of a list with the first item reduced in size, so that we can conclude that NF is strongly monotonic.

That NFD is not strongly monotonic follows from the above lists; $\mathcal{L}' \leqslant_{\mathrm{D}} \mathcal{L}$ and $NFD(\mathcal{L}') > NFD(\mathcal{L})$.

[1]We note that the lists (E13) and (E14) show that no ∞-space, on-line, conservative bin-packing heuristic is monotonic, and that the same holds for off-line heuristics that sort the list into decreasing order before applying an on-line heuristic.

We note further that these lists provide a lower bound of $\frac{1}{12}$ for the asymptotic worst-case nonmonotonicity, as defined by Murgolo.[50] This [new] lower bound improves upon all the known lower bounds; see [14, Theorem 2.16] for an up-to-date overview of these bounds.

β	nr of cases	β	nr of cases
1	–	11	110880
2	2	12	221760
3	6	13	2882880
4	12	14	5765760
5	60	15	17297280
6	120	16	34594560
7	840	17	588107520
8	1680	18	1176215040
9	5040	19	22348085760
10	10080	20	44696171520

Table E.3. Number of harmonic CSPs to be investigated for critical item $\geqslant \frac{1}{\beta}$

E.6 Case enumeration for the parametric harmonic CSP

The following algorithm[2] was used to determine the parametric gap, $\gamma_{\mathrm{h}}((\beta))$ for the harmonic CSP with smallest item $1/\beta$. It was run for the values $2 \leqslant \beta \leqslant 11$.

The idea is to certify as many candidate lists as possible, either as having a gap strictly less than 1 or as having a gap of at least 1. The ones with a gap < 1 are of no interest and of the ones with a gap $\geqslant 1$ only the largest one is of interest. This leaves a relatively small set of instances that still need to be certified.

Algorithm Generate all harmonic lists with $f_i \leqslant \mathrm{sp}(i) - 1$ and $2 \leqslant i \leqslant \beta$, and perform the following checks for each list $\mathcal{L}$. Denote by γ^* the maximum gap found so far by the algorithm.

1. $(Mat \bmod 1) = 0 \ \Rightarrow$ discard the list since $\gamma = 0$ or 1
2. $(Mat \bmod 1) \geqslant 2 - \gamma^* \ \Rightarrow$ discard the list since $\gamma \leqslant \gamma^*$
3. $\lceil Mat \rceil = 1 + \lceil \frac{Mat-1}{1-1/\beta} \rceil \ \Rightarrow$ discard the list since $\gamma < 1$
4. $\lceil Mat \rceil = FFD \ \Rightarrow$ discard the list since $\gamma < 1$
5. Determine $\varphi = \max\{\sum a_i/i \mid a_i \in \mathbb{N}, a_i \leqslant f_i\}$
   (a) $\varphi = 1$: discard the list, since there is a smaller instance with the same or larger gap.
   (b) $\lceil \frac{Mat}{\varphi} \rceil > \lceil Mat \rceil \ \Rightarrow \gamma > 1$: keep the list if $\gamma > \gamma^*$ and update γ^*, otherwise discard the list.
6. Determine a heuristic solution H, by repeatedly solving a subset-sum problem.
   If $H = \lceil Mat \rceil$ then discard list, otherwise output list to be analysed subsequently.

At the end of the algorithm we have a list with gap γ^* and a set of candidate lists which need to be analysed further[3] to determine if any of them has a gap $> \gamma^*$.

[2]The algorithm was implemented in integer arithmetic to avoid rounding- and truncation errors, which otherwise might lead to an incorrect certification.
[3]It turns out that for $\beta \leqslant 11$ this set is always empty.

	2	3	4	5	6	7	8	9	10	*Mat*	γ_h
1	1	2	0	4						1.967	1.03333
2	0	2	1	1	0	6				1.974	1.02619
3	0	2	0	3	0	5				1.981	1.01905
4	1	2	1	0	0	4				1.988	1.01190
5	0	1	0	4	0	6				1.990	1.00952
6	1	2	0	2	0	3				1.995	1.00476
7	1	1	1	1	0	5				1.998	1.00238
8	0	2	1	1	1	5				1.998	1.00238
9	0	1	0	4	0	5	1			1.973	1.02738
10	1	1	0	3	0	3	1			1.987	1.01310
11	1	0	1	2	0	5	1			1.989	1.01071
12	1	0	0	4	0	4	1			1.996	1.00357
13	0	1	0	4	1	4	1			1.996	1.00357
14	1	2	1	3	0	6	1			2.999	1.00119
15	0	0	1	3	1	6	1			1.999	1.00119
16	1	1	0	4	0	0	1	2		1.981	1.01944
17	1	2	1	1	0	1	0	2		1.982	1.01825
18	1	2	1	3	0	6	0	1		2.985	1.01508
19	1	2	0	3	0	0	0	2		1.989	1.01111
20	1	1	1	2	0	2	0	2		1.991	1.00873
21	0	2	1	2	1	2	0	2		1.991	1.00873
22	1	2	0	0	0	5	0	1		1.992	1.00794
23	1	2	1	4	0	3	1	2		2.992	1.00754
24	1	0	0	2	0	6	1	1		1.993	1.00675
25	1	1	1	4	0	0	0	1		1.994	1.00556
26	0	2	1	4	1	0	0	1		1.994	1.00556
27	1	2	1	1	0	1	1	1		1.996	1.00437
28	1	0	1	0	1	6	0	2		1.996	1.00397
29	0	2	1	0	0	6	0	2		1.996	1.00397
30	1	1	0	4	0	1	0	2		1.998	1.00159
31	0	2	0	4	1	1	0	2		1.998	1.00159
32	1	0	1	2	1	4	0	1		1.999	1.00079
33	0	2	1	2	0	4	0	1		1.999	1.00079
34	1	2	0	1	0	2	1	2		2.000	1.00040
35	1	2	0	0	0	5	0	0	1	1.981	1.01905
36	0	2	1	0	0	6	0	1	1	1.985	1.01508
37	0	2	1	2	0	4	0	0	1	1.988	1.01190
38	1	1	0	1	0	6	0	0	1	1.990	1.00952
39	0	2	0	1	1	6	0	0	1	1.990	1.00952
40	0	2	0	2	0	5	0	1	1	1.992	1.00794
41	0	2	0	4	0	3	0	0	1	1.995	1.00476
42	0	2	1	3	0	1	1	1	1	1.996	1.00437
43	0	0	1	2	1	6	0	2	1	1.996	1.00397
44	0	1	1	3	0	5	0	0	1	1.998	1.00238
45	1	0	1	0	1	6	1	0	1	1.999	1.00119
46	0	2	1	0	0	6	1	0	1	1.999	1.00119
47	0	0	1	4	1	4	0	1	1	1.999	1.00079
48	0	2	0	3	0	2	1	2	1	2.000	1.00040

Table E.4. Harmonic CSPs with $\gamma_h > 1$
Listed are all instances of the harmonic CSP, which have a gap $\gamma > 1$ and a smallest, critical item $\leqslant 1/10$.

E.7 Worst-case ratios

In this section we will derive some lemmas that show the relationship between 'minimal' bounds for bin-packing heuristics, and their asymptotic and recurrent ratio. In analogy with the notation for the asymptotic behaviour of univariate functions, we define the following to facilitate the notation. If $f(\mathcal{L})$ and $g(\mathcal{L})$ are two functions defined on lists, then $f(\mathcal{L}) = o(g(\mathcal{L}))$ means $\lim_{z\to\infty} \sup_{\{\mathcal{L}|g(\mathcal{L})\geqslant z\}} f(\mathcal{L})/g(\mathcal{L}) = 0$.

E14 **Lemma** *If there exist constants c and r such that the bound* $\forall\mathcal{L}\ H(\mathcal{L}) \leqslant c + rOPT(\mathcal{L})$ *holds, and r is the least such, then* $r = \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})} \equiv R_H^\infty$

Proof. *A lower bound for r is given by the following.*

$$\forall\mathcal{L}\ r \geqslant \frac{H(\mathcal{L})-c}{OPT(\mathcal{L})} \;\Rightarrow\; \forall z\ r \geqslant \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})-c}{OPT(\mathcal{L})} \;\Rightarrow\; r \geqslant \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})},$$

where the first inequality is just a rewrite of the bound, the second one follows by conditioning on the value of OPT*, and the last one by letting* $z \to \infty$*, so that the constant vanishes. If there is an* $\varepsilon > 0$ *and* z *such that* $H(\mathcal{L}) \leqslant \tilde{c} + (r-\varepsilon)OPT(\mathcal{L})$ *holds for all lists with* $OPT(\mathcal{L}) \geqslant z$*, for some constant* $\tilde{c}$*, then* r *cannot be minimal, so that the minimality of r implies the following.*

$$\forall\varepsilon > 0\ \forall z\ \exists\{\mathcal{L} \mid OPT(\mathcal{L}) \geqslant z\} \qquad H(\mathcal{L}) > c + (r-\varepsilon)OPT(\mathcal{L})$$

An upper bound can now be proven as follows:

$$\begin{aligned}
\forall\varepsilon > 0\ \forall z\ \exists\{\mathcal{L} \mid OPT(\mathcal{L}) \geqslant z\} \quad & r < \varepsilon + \frac{H(\mathcal{L})-c}{OPT(\mathcal{L})},\\
\forall\varepsilon > 0\ \forall z \quad & r < \varepsilon + \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})-c}{OPT(\mathcal{L})},\\
\forall\varepsilon > 0\ \forall z \geqslant -c/\varepsilon \quad & r < 2\varepsilon + \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})},\\
\forall\varepsilon > 0 \quad & r < 2\varepsilon + \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})},\\
& r \leqslant \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{H(\mathcal{L})}{OPT(\mathcal{L})}
\end{aligned}$$

The first inequality is a rewrite of the minimality condition, the second one relaxes the RHS of the bound, for the third we choose $z \geqslant -c/\varepsilon$ *to eliminate the constant c, and the fourth one is derived by letting* $z \to \infty$*. The last inequality then follows directly. Combining the lower and upper bound proves the lemma.* □

E15 **Lemma** *If* $r = R_H^\infty < \infty$ *and* $c(\mathcal{L}) = H(\mathcal{L}) - rOPT(\mathcal{L})$*, then* $c(\mathcal{L}) = o(OPT(\mathcal{L}))$

Proof. *The definition of the asymptotic ratio as a limsup gives the following two inequalities.*

$$\forall\varepsilon > 0\ \exists z\ \forall\{\mathcal{L} \mid OPT(\mathcal{L}) \geqslant z\}\ \frac{H(\mathcal{L})}{OPT(\mathcal{L})} \leqslant r + \varepsilon \;\Rightarrow\; \forall\varepsilon > 0\ \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{c(\mathcal{L})}{OPT(\mathcal{L})} \leqslant \varepsilon$$

$$\forall\varepsilon > 0\ \forall z\ \exists\{\mathcal{L} \mid OPT(\mathcal{L}) \geqslant z\}\ \frac{H(\mathcal{L})}{OPT(\mathcal{L})} > r - \varepsilon \;\Rightarrow\; \forall\varepsilon > 0\ \lim_{z\to\infty} \sup_{\{\mathcal{L}|OPT(\mathcal{L})\geqslant z\}} \frac{c(\mathcal{L})}{OPT(\mathcal{L})} \geqslant -\varepsilon$$

Substitution of $H = c + rOPT$ *and letting* $z \to \infty$ *gives the second inequalities as a consequence. Letting* $\varepsilon \to 0$ *now proves the lemma.* □

E16 Lemma *If there exist constants c and r such that the bound* $\forall k \in \mathrm{N}^+ \;\; H(k\mathcal{L}) \leqslant c + r\, CSP_R(k\mathcal{L})$ *holds, and r is the least such, then* $r = \limsup_{k\to\infty} \frac{H(k\mathcal{L})}{k\, CSP_R(\mathcal{L})} \equiv R_H^{rec}(\mathcal{L})$

Proof. *A lower bound for r is given by the following.*

$$\forall k \;\; r \geqslant \tfrac{H(k\mathcal{L})-c}{CSP_R(k\mathcal{L})} \quad \Rightarrow \quad r \geqslant \limsup_{k\to\infty} \tfrac{H(k\mathcal{L})}{CSP_R(k\mathcal{L})},$$

where the first inequality is just a rewrite of the bound, and the second inequality follows by letting $k \to \infty$. *If there is an* ε *and* k_0 *such that* $H(k\mathcal{L}) \leqslant \tilde{c} + (r-\varepsilon)\, CSP_R(k\mathcal{L})$ *holds for all* $k \geqslant k_0$, *for some constant* $\tilde{c}$, *then r cannot be minimal, so that the minimality of r implies the following.*

$$\forall \varepsilon > 0 \;\; \forall k_0 \;\; \exists k \geqslant k_0 \qquad H(k\mathcal{L}) > c + (r-\varepsilon)\, CSP_R(k\mathcal{L})$$

An upper bound can now be proven as follows, where $z = CSP_R(\mathcal{L})$ *is substituted for notational brevity,*

$$\begin{aligned}
\forall \varepsilon > 0 \;\; \forall k_0 \;\; \exists k \geqslant k_0 \qquad & r < \varepsilon + \tfrac{H(k\mathcal{L})-c}{kz},\\
\forall \varepsilon > 0 \;\; \forall k_0 \qquad & r < \varepsilon + \sup_{k\geqslant k_0} \tfrac{H(k\mathcal{L})-c}{kz},\\
\forall \varepsilon > 0 \;\; \forall k_0 \geqslant \tfrac{-c}{\varepsilon z} \qquad & r < 2\varepsilon + \sup_{k\geqslant k_0} \tfrac{H(k\mathcal{L})}{kz},\\
\forall \varepsilon > 0 \qquad & r < 2\varepsilon + \limsup_{k\to\infty} \tfrac{H(k\mathcal{L})}{kz},
\end{aligned}$$

where the first inequality is a rewrite of the minimality condition, the second one relaxes the RHS of the bound, for the third we choose $k_0 \geqslant \frac{-c}{\varepsilon z}$ *to eliminate the constant c, and the fourth one is derived by letting* $z \to \infty$ *for* ε *fixed. The last inequality now follows directly. Combining the lower and upper bound proves the lemma.* □

E17 Lemma *If* $r = R_H^{rec}(\mathcal{L}) < \infty$ *and* $c(k) = H(k\mathcal{L}) - r\, CSP_R(k\mathcal{L})$, *then* $c(k) = o(k)$

Proof. *The definition of the recurrent ratio as a limsup gives the following two inequalities.*

$$\begin{aligned}
&\forall \varepsilon > 0 \;\; \exists k_0 \;\; \forall k \geqslant k_0 \;\; \tfrac{H(k\mathcal{L})}{CSP_R(k\mathcal{L})} \leqslant r + \varepsilon \quad \Rightarrow \quad \forall \varepsilon > 0 \;\; \limsup_{k\to\infty} \tfrac{c(k)}{CSP_R(k\mathcal{L})} \leqslant \varepsilon\\
&\forall \varepsilon > 0 \;\; \forall k_0 \;\; \exists k \geqslant k_0 \;\; \tfrac{H(k\mathcal{L})}{CSP_R(k\mathcal{L})} > r - \varepsilon \quad \Rightarrow \quad \forall \varepsilon > 0 \;\; \limsup_{k\to\infty} \tfrac{c(k)}{CSP_R(k\mathcal{L})} > -\varepsilon
\end{aligned}$$

Substitution of $H = c + r\, CSP_R$ *and letting* $k \to \infty$ *gives the second inequalities. Let* $\varepsilon \to 0$ *and recall that* $CSP_R(k\mathcal{L}) = k\, CSP_R(\mathcal{L})$ *and that* $CSP_R(\mathcal{L})$ *is a constant. This shows that* $\limsup_{k\to\infty} c(k)/k = 0$ *and proves the lemma.* □

References

[1] A. V. Aho and N. J. A. Sloane. Some doubly exponential sequences. *Fibonacci Quarterly*, 11:429–437, 1973.

[2] C. Aldridge, J. Chapman, R. Gower, R. Leese, C. McDiarmid, M. Shepherd, H. Tuenter, H. Wilson, and A. Zinober. Pattern reduction in paper cutting. In *29th European Study Group with Industry, Oxford, 18–22 March 1996*, Mathematical Institute, 1996. University of Oxford.

[3] Brenda S. Baker. A new proof for the first-fit decreasing bin-packing algorithm. *Journal of Algorithms*, 6:49–70, 1985.

[4] Brenda S. Baker and E. G. Coffman, Jr. A tight asymptotic bound for next-fit-decreasing bin-packing. *SIAM Journal on Algebraic and Discrete Methods*, 2(2):147–152, June 1981.

[5] S. Baum and L.E. Trotter, Jr. Integer rounding for polymatroid and branching optimization problems. *SIAM Journal on Algebraic and Discrete Methods*, 2(4):416–425, December 1981.

[6] E. T. Bell. *Men of Mathematics.* Simon & Schuster, New York, 1986.

[7] A. R. Brown. *Optimum Packing and Depletion: The computer in space- and resource-usage problems.* Number 14 in Computer Monographs. Macdonald, London, 1971.

[8] Bernard Chazelle. The bottom-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, C-32(8):697–707, August 1983.

[9] C. H. Cheng, B. R. Feiring, and T. C. E. Cheng. The cutting stock problem – a survey. *International Journal of Production Economics*, 36:291–305, 1994.

[10] Vašek Chvátal. *Linear Programming.* W. H. Freeman and Company, New York, 1983. In particular Chapter 13; "The Cutting Stock Problem": pp. 195–212.

[11] E. G. Coffman, Jr. An introduction to proof techniques for bin-packing approximation algorithms. In Dempster et al.,[18] pages 245–270.

[12] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin-packing algorithms: An updated survey. In G. Ausiello, M. Lucertini, and A. Serafini, editors, *Algorithm Design*

for Computer System Design, number 284 in CISM Courses and Lectures, pages 49–106. Springer Verlag, Wien–New York, 1984.

[13] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Bin packing with divisible item sizes. *Journal of Complexity*, 3:406–428, 1987.

[14] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 2. PWS Publishing Company, Boston, MA, 1997. (in press).

[15] E. G. Coffman, Jr. and Georg S. Lueker. *Probabilistic Analysis of Packing and Partioning Algorithms.* Interscience Series in Discrete Mathematics and Optimization. J. Wiley and Sons, Inc, New York, 1991.

[16] Don Coppersmith and Prabhakar Raghavan. Multidimensional on-line bin packing: Algorithms and worst-case analysis. *Operations Research Letters*, 8(1):17–20, February 1989.

[17] J. Csirik. The parametric behavior of the first-fit decreasing bin packing algorithm. *Journal of Algorithms*, 15:1–28, 1993.

[18] M.A.H. Dempster, J. K. Lenstra, and A. H. G. Rinnooy Kan, editors. *Deterministic and Stochastic Scheduling.* D. Reidel Publishing Company, Dordrecht, Holland, 1982. Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, Durham, England, July 6–17, 1981.

[19] Adolf Diegel. Integer LP solution for large trim problems. Paper presented at the EURO IX, TIMS XXVIII conference, Paris, France, July 1988.

[20] Harald Dyckhoff and Ute Finke. *Cutting and Packing in Production and Distribution: A Typology and Bibliography.* Physica–Verlag, Heidelberg, 1992.

[21] K. Eisemann. The trim problem. *Management Science*, 3(3):279–284, April 1957.

[22] William and Fern Ellison. *Prime Numbers.* Wiley Interscience Publication. John Wiley & Sons, Inc., New York, 1985.

[23] *Encyclopaedia of Mathematics.* Kluwer Academic Publishers, Dordrecht, 1989.

[24] Martin Fieldhouse. The duality gap in trim problems. *SICUP–Bulletin*, 5, November 1990.

[25] Martin Fieldhouse. The duality gap problem: Solved? Paper presented at the MUGI 21 conference, Budapest, Hungary, October 1991.

[26] Martin Fieldhouse. The duality gap problem in trim problems: Not yet solved. working paper, 1995.

[27] Marshall L. Fisher. Worst-case analysis of heuristic algorithms for scheduling and packing. In Dempster et al.,[18] pages 15–34.

[28] Gábor Galambos and Gerhard J. Woeginger. On-line bin packing – a survey. Report 280, Institut für Mathematik, Technische Universität Graz, Steyrergasse 30, A-8010 Graz, Austria, December 1993.

[29] Martin Gardner. Mathematical games: Some packing problems that cannot be solved by sitting on the suitcase. *Scientific American*, pages 22–26, October 1979. (see also November issue, p. 29).

[30] M. R. Garey, R. L. Graham, D. S. Johnson, and A. C. Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory (A)*, 21:257–298, 1976.

[31] Michael R. Garey and David S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[32] T. Gau. Counterexamples to the IRU property. *SICUP-Bulletin*, 12, November 1994.

[33] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, November–December 1961.

[34] F. Göbel. Geometrical packing and covering problems. In Schrijver,[62] pages 179–199.

[35] Solomon W. Golomb. On certain non–linear recurring sequences. *The American Mathematical Monthly*, 7:403–405, 1963.

[36] M. Grötschel and L. Lovász. Combinatorial optimization. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, chapter 28, pages 1541–1576. Elsevier Science B.V., 1995. In particular p. 1551 (Next-Fit) and p. 1574 (First-Fit).

[37] D. S. Johnson, A. Demers, J. D. Ullman, M. R. Garey, and R. L. Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal of Computing*, 3(4):299–325, December 1974.

[38] David S. Johnson. *Near-Optimal Bin Packing Algorithms*. Ph.D thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1973.

[39] David S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8:272–314, 1974.

[40] David S. Johnson. Private communication, June 1996.

[41] David S. Johnson and Michael R. Garey. A 71/60 theorem for bin packing. *Journal of Complexity*, 1:65–106, 1985.

[42] Michael A. Langston. A study of composite heuristic algorithms. *Journal of the Operational Research Society*, 38(6):539–544, 1987.

[43] C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. *Journal of the Association for Computing Machinery*, 32(3):562–572, July 1985.

[44] Frank M. Liang. A lower bound for on-line bin packing. *Information Processing Letters*, 10(2):76–79, March 1980.

[45] Weizhen Mao. *Scheduling and Bin Packing: A Study of the Worst-Case Performance Bounds.* Ph.D. thesis, Department of Computer Science, Princeton University, Princeton, NJ, 1990.

[46] Weizhen Mao. Tight worst-case performance bounds for next-k-fit bin packing. *SIAM Journal of Computing*, 22(1):46–56, February 1993.

[47] Odille Marcotte. The cutting stock problem and integer rounding. *Mathematical Programming*, 33:82–92, 1985.

[48] Odille Marcotte. An instance of the cutting stock problem for which the rounding property does not hold. *Operations Research Letters*, 4(5):239–243, February 1986.

[49] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley and Sons, 1990.

[50] Frank D. Murgolo. Anomalous behavior in bin packing algorithms. *Discrete and Applied Mathematics*, 21:229–243, 1988.

[51] M. Nair. On Chebyshev–type inequalities for primes. *The American Mathematical Monthly*, 89(2):126–129, February 1982.

[52] Hoon Liong Ong, M.J. Magazine, and T.S. Wee. Probabilistic analysis of bin packing heuristics. *Operations Research*, 32(5):983–998, Sep–Oct 1984.

[53] A. E. Paull and John R. Walter. The trim problem: An application of linear programming to the manufacture of newsprint paper. *Econometrica*, 23(3):336, July 1955. Abstract of paper presented at annual meeting of the Econometrica Society in Montreal, Canada, September 10–13, 1954.

[54] John F. Pierce, Jr. *Some Large-Scale Production Scheduling Problems in the Paper Industry.* The Ford Foundation Doctoral Dissertation Series. Prentice–Hall, Englewood Cliffs, N.J., 1964.

[55] Prakash Ramanan. Average-case analysis of the smart next fit algorithm. *Information Processing Letters*, 31(5):221–225, June 1989.

[56] WanSoo T. Rhee. Probabilistic analysis of the next fit decreasing algorithm for bin-packing. *Operations Research Letters*, 6(4):189–191, September 1987.

[57] Paulo Ribenboim. *The Book of Prime Number Records.* Springer Verlag, Heidelberg, second edition, 1989.

[58] Hans Riesel. *Prime Numbers and Computer Methods for Factorization.*, volume 57 of *Progress in Mathematics.* Birkhäuser, Boston, Mass., 1985.

[59] Gary M. Roodman. Near-optimal solutions to one-dimensional cutting stock problems. *Computers & Operations Research*, 13(6):713–719, 1986.

[60] Guntram Scheithauer and Johannes Terno. A remark to the duality gap. *SICUP–Bulletin*, 6, June 1991.

[61] Guntram Scheithauer and Johannes Terno. About the gap between the optimal values of the integer and continuous relaxation one-dimensional cutting stock problem. In *DEGOR–Proceedings* 1991, pages 439–444, Berlin Heidelberg, 1992. Springer Verlag.

[62] A. Schrijver, editor. *Packing and Covering in Combinatorics*, volume 106 of *Mathematical Centre Tracts*. Mathematical Centre, Tweede Boerhaavestraat 49, Amsterdam, 1979. Proceedings of the study week "Stapelen en Overdekken" (Packing and Covering), June 5–9, 1978.

[63] SICUP-Bulletin, Newsletter of the Special Interest Group on Cutting and Packing. Available via anonymous ftp: sicup@or1.nat.tu–bs.de.

[64] David Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Research Logistics*, 41:579–585, 1994.

[65] Hartmut Städtler. A one-dimensional cutting stock problem in the aluminium industry and its solution. Paper presented at the EURO IX, TIMS XXVIII conference, 6–8 July 1988, Paris, France (published in EJOR 44–2).

[66] Paul E. Sweeney and Elizabeth L. Ridenour. Cutting and packing problems: A categorized application-oriented research bibliography. Working Paper 610, The University of Michigan, School of Business Administration, Ann Arbor, Michigan, 1989. (published in EJOR 43–7).

[67] Maciej M. Syslo, Narsingh Deo, and Janusz S. Kowalik. *Discrete Optimization Algorithms with PASCAL programs*. Prentice–Hall, Inc, 1983.

[68] Jan van Tiel. *Convex Analysis: An Introductory Text*. John Wiley and Sons, Chicester, 1984.

[69] Hans J. H. Tuenter. Towards on-line allocation or the cutting stock problem in the carpet industry. Master's thesis, Faculty of Applied Mathematics, University of Twente, Enschede, The Netherlands, June 1991.

[70] Kaihong Xu. *A Bin-Packing Problem with Item Sizes in the Interval* $(0, \alpha]$ *for* $\alpha \leqslant \frac{1}{2}$. Ph.D. thesis, Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, 1993.

[71] Andrew Chi-Chih Yao. New algorithms for bin packing. *Journal of the Association for Computing Machinery*, 27(2):207–227, April 1980.

[72] Min-Yi Yue. A simple proof of the inequality $\mathrm{FFD}(L) \leqslant \frac{11}{9}\mathrm{OPT}(L) + 1$, $\forall L$ for the FFD bin-packing algorithm. *Acta Mathematicae Applicatae Sinica*, 7(4):321–331, October 1991.

Glossary of Symbols

Index