Packing Shelves with Divisible Items

Moshe Dror *
James B. Orlin †
Michael Zhu‡

September 9, 2004

Abstract

In this paper we examine the issue of how likely it is that one can pack two shelves of integer length L by items whose individual lengths are divisors of L, given that the individual items sum-up to $2L$. The main thrust of this study is computational. That is, we explicitly compute the answer for this question for all integers $L, 1 \leq L \leq 1000$. We conclude that an instance of this packing problem in which we cannot pack the two shelves is very rare (see Tables 1-4). Existence of packing failures is tied to the number of divisors of L. That is, we prove that the number of divisors has to be at least 8 for a packing failure to exist.

*MIS Department, College of Business and Public Administration, University of Arizona, Tucson, Arizona 85721, and Operations Research Center, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, U.S.A.

†Sloan School of Management, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, U.S.A.

‡Sloan School of Management, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, U.S.A.

1 Introduction

Consider the following: You are given a one dimensional shelf with integer length, and a set one dimensional items also of integer length. Packing the items onto the shelf without overlay is possible if and only if the sum of the items' lengths is less or equal to the length of the shelf. On the other hand, if you are given two identical shelves it is no longer clear if the items will fit into the shelves with just the condition that the sum of their lengths is less than or equal to twice the length of a shelf.

The understanding of when we can assure the fitting of items onto a shelf space has a number of interesting practical implications. In particular, this type of question comes up in computer technology (VLSI) in telecommunications industry where a number of different modular components (switches, transmitters, etc.) may be plugged in, replaced, or added into "shelf slots" in an equipment bay. One may think of other examples such as shelf space in supermarkets packed with different cereal or detergent boxes.

More formally, denote by Z^+ the set of positive integers and let $L \in Z^+$ denote the shelf length (number of slots). Let $U = \{u_1, \ldots, u_n\}$ be the set of different items where $v_i(u_i) \in Z^+$ denotes the length of item $u_i, i = 1, \ldots, n$. We require $v_i(u_i) \neq v_j(u_j)$ for $i \neq j$. Let $a_i \in Z^+$ denote the number of copies we are given (≥ 1) of item $u_i, i = 1, \ldots, n$. With each item $u_i \in U$ we associate a tupple (v_i, a_i) denoting the length and the "frequency" (the number of copies) of item u_i. With a minimal risk of confusion, we mostly refer to item u_i by its length v_i.

If we have only one shelf, then we know that we can pack (shelve) all the items in U if and only if $\sum_{i=1}^{n} a_i v_i \leq L$. If we have two shelves and the condition that $\sum_{i=1}^{n} a_i v_i \leq 2L$, can we still shelve the items in U ? We are particularly interested to know the answer in the case when the $\sum_{i=1}^{n} a_i v_i = 2L$, or in the case of $m \geq 2$ shelves, when the $\sum_{i=1}^{n} a_i v_i = mL$. That is, when the shelves have to be completely shelved – filled up.

With no restrictions on the length values $v_i, i = 1, \ldots, n$, this problem is simply referred to as the NUMBER PARTITION PROBLEM (see Garey and Johnson, 1979) when $m = 2$, or m-PARTITION for $m > 2$. More precisely, the Number Partition Problem is described as follows:

NUMBER PARTITION PROBLEM.
INPUT: Nonnegative integers $w_1, \ldots, w_n$ (not necessarily all different).

QUESTION: Is there a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = \frac{1}{2} \sum_{i=1}^{n} w_i$?

If the answer is Yes, we say that the set of integers $w_1, \ldots, w_n$ is *partitionable*. Otherwise, we say that it is *non-partitionable*.

It is well known that the Number Partition Problem is NP-complete, and that it can be solved in pseudo-polynomial time; that is, the running time is polynomial in n and $w_{\max} = \max\{w_i : 1 \leq i \leq n\}$.

We now consider a generalization of the Number Partition Problem.

UNIVERSAL NUMBER PARTITION PROBLEM.
INPUT: Integers $w_1, \ldots, w_n, b$.

QUESTION: Is it true that for every subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = 2b$, the set $\{w_i : i \in S\}$ is partitionable ?

The Number Partition Problem is a special case of the Universal Number Partition Problem is which $\sum_{i=1}^{n} w_i = 2b$. The complement of the Universal Number Partition Problem is also naturally stated in the context of computational complexity theory. The complementary problem is to determine whether there is a subset S with $\sum_{i \in S} w_i = 2b$ and such that the set $\{w_i : i \in S\}$ is non-partitionable.

The Universal Number Partition Problem is NP-hard. However, it might not be in the class NP-equivalent (see Garey and Johnson, 1979). The complement of the Universal Number Partition Problem is in the class $\sum_2^P$, or as Garey and Johnson (1979) refer to, in the class NP^{NP}.

In this paper, we consider the special case of the Universal Number Partition Problem in which each w_i is a divisor of b, and in which there are $\frac{2b}{w_i}$ copies of integer w_i. We refer to this special case as the *Universal Shelf Packing Problem* (USPP).

We say a collection S of integers $\{w_1, \ldots, w_n\}$ is a *candidate* for L if w_i is a divisor of L for $1 \leq i \leq n$, and if $\sum_{i=1}^{n} w_i = 2L$. We say that a candidate is a *success* if it is partitionable. We say that the shelf length L is *universally packable* if every candidate is a success. If L is not universally packable, then there is a candidate for L that is a *failure*, that is, it is not partitionable.

Getting back to the motivation for examining the USPP, suppose that one is given a number of different modular components that may be plugged in, replaced, or added into "shelf slots" in an equipment bay – a set of shelves. Assuming that the total length of the modular components is $2L$ with each component's length a divisor of L, can the components be packed into two shelves ? More generally, one can address the related question when the total length of the components is between mL and $(m + 1)L$, can the components be packed into $(m + 1)$ shelves. We address the theoretical issue of how one determines when L is universally packable and when L has failures.

Even though the Universal Number Partition Problem is NP-hard (and possibly harder than any problem in the class NP), the complexity of the USPP is not known. This shelving problem was defined and analyzed by Dror et al. (1994). In fact, Dror et al. have provided a set of conditions, in addition to divisibility of L by the v_i's, which assure that the shelves can be fully shelved. These additional conditions are in the form of a minimal set of sufficiency conditions. However for many instances of the shelf-packing problem the sufficiency conditions stated in Dror et al. (1994) are not necessary for a packing (shelving) solution to exist. In a great majority of cases, just the divisibility of L by the corresponding v_i's would allow for the shelves to be completely packed. This was exactly the argument put forward by a group of telecommunication engineers who faced this problem in practice. The engineers were 'almost' right. This paper computes the likelihood of being able to fully pack two shelves of length L, given that $\sum_{i=1}^{n} a_i v_i = 2L$, and $v_i | L, \ i = 1, \ldots, n$, with no other conditions attached. In other words, we calculate for L with $1 \leq L \leq 1000$ how confident the engineers can be in their argument that the two shelves can be fully shelved.

In this paper we provide additional theoretical results concerning the shelving problem.

The primary results of this paper are as follows:

1. If L has fewer than 8 divisors, then L is universally packable. We conjecture that L is universally packable whenever L has at most two prime divisors.

2. We show that if L has a failure, it has a failing candidate solution S of integers with the property that if kw_i is a divisor of L, then there are fewer than k copies of integer w_i in S.

3. We give algorithms for the following:
   a. For a given value of L, we count the number of different candidate sets.
   b. For a given value of L, we count the number of candidate sets satisfying the bound given in (2).
   c. We give a dynamic program for determining whether a candidate set is a success or failure.

4. We enumerated all distinct failures for L with $1 \leq L \leq 1000$. The number of values L with failures is 217. We conjecture that the percentage of integers L that are universally packable goes to zero as L goes to infinity.

We start with a number of examples. Some of the examples are taken from Dror et al. (1994). We list the items in a nonincreasing order of their length. (With some risk for confusion we use interchangeably the words *packing* and *shelving* to mean the same thing.)

Example 1: $L = 30$ (shelf size), $m = 3$ (the number of shelves), $n = 4$ (the number of different v_i's.

v_i	15	10	6	3
a_i	1	3	5	5

The total shelf length is $90 = 3 \times 30 = 1 \times 15 + 3 \times 10 + 5 \times 6 + 5 \times 3$.

It is easy to see in this case a packing solution: In the first shelf we can simply pack the first item (length 15) with two items of length 6 each and one item of length 3. The second shelf can be packed with three items of length 10 each, and the third shelf with three items of length 6 and four items of length 3.

Divisibility $v_i|L$ is important for packing as can be seen in the example below.

Example 2: $L = 20, m = 3, n = 4$**.**

v_i	15	10	6	3
a_i	1	3	2	1

The total shelf length is $60 = 3 \times 20 = 1 \times 15 + 3 \times 10 + 2 \times 6 + 1 \times 3$, but after packing item 1 of length 15, no remaining component combination can fill the remaining shelf space of 5.

Example 3: $L = 60, m = 2, n = 4$**.** Divisibility alone does not necessarily imply the existence of a packing solution.

v_i	30	20	12	2
a_i	1	2	4	1

Again, $\sum_{i=1}^{4} a_i v_i = 2 \times 60 = 120$. The lengths $v_i, i = 1, \ldots, 4$, all divide L. However, we cannot pack the two shelves.

Example 4: $L = 30, m = 2, n = 4$**.** The smallest L for which packing might fail even when all v_is divide L is demonstrated in this example.

v_i	15	10	6	1
a_i	1	2	4	1

As in Example 3, the sum of the items' lengths equals the shelves' length and the lengths $v_i, i = 1, \ldots, 4$, all divide L, but we cannot pack the shelves.

Example 5: $L = 30, m = 2, n = 3$**.**

v_i	15	6	3
a_i	1	2	9

It is easy to see that the two shelves can be fully packed in this case.

2 Sufficiency Conditions

In the five examples above we presented two instances where we can shelve the items and three instances where full packing is impossible. Given a shelf-packing problem instance, we would like to know if we can or cannot shelve the items. Before stating the conditions which assure the existence of a packing solution we restate the notation used in this paper.

Suppose a set of items $U = \{u_1, \ldots, u_n\}$ together with their length values $\{v_1, \ldots, v_n\}$, and a shelf of length L. All the participating values are positive integers. That is, $L \in Z^+$, and $v_i \in Z^+, i = 1, \ldots, n$. Let the set of positive integers $H = \{h_1, \ldots, h_K\},\ h_k \in Z^+, k = 1, \ldots, K,\ h_i \neq h_j$ for $i \neq j$, be such that $L = \Pi_{k=1}^{K} h_k$. Let each $v_i = \Pi_{h_j \in H_i} h_j$, for some $H_i \subseteq H$. Denote the divisors h_j common to all v_i's ($i = 1, \ldots, n$) by $\bar{H} = \cap_{i=1}^{n} H_i$. Note that in this description $v_i | L, i = 1, \ldots, n$. In addition, we are given a set $A = \{a_1, \ldots, a_n\}, a_i \in Z^+, i = 1, \ldots, n$, representing the number of copies (the frequency) of the corresponding items, an $m \in Z^+$, and the relation $\sum_{i=1}^{n} a_i v_i = mL$.

Given an instance of a packing problem as above, the subset $\bar{H}$ may be empty. However, if beside the divisors in $\bar{H}$, no pair of H_i, H_j subsets share any other divisors (That is, $\{H_i \backslash \bar{H}\} \cap \{H_j \backslash \bar{H}\} = \emptyset$, for all pairs i and j ($i \neq j$).), we obtain an interesting result.

Theorem 1: (Dror et al. 1994) If $\{H_i \setminus \bar{H}\} \cap \{H_j \setminus \bar{H}\} = \emptyset$, for all pairs i and j ($i \neq j$), then the shelf-packing problem has a solution – can be fully packed.

We do not repeat here the proof from Dror et al. (1994). The existence of the sets H and $H_i, i = 1, \ldots, n$ as above, is assured by prime factorization. We could either state explicitly the divisibility property of L by each v_i, or express it in terms of the sets H, and H_i respectively.

Theorem 1 states sufficiency conditions for the packing shelves solution. Any relaxation of the requirements in Theorem 1 results in an existence of packing failure instance, which is demonstrated by Examples 2,3,4, and 5.

A mathematical program (a system of nonnegative linear diophantine equations) of the packing problem can be stated as follows:

$$\sum_{i=1}^{n} v_i x_{ij} = L, \; j = 1, \ldots, m \tag{1}$$

$$\sum_{j=1}^{m} x_{ij} = a_i, \; i = 1, \ldots, n \tag{2}$$

where x_{ij} is a nonnegative integral solution to (1) and (2), representing the number of items u_i shelved in shelf j. Thus, Theorem 1 states sufficiency conditions for a nonnegative integral solution to (1) and (2) (see Schrijver, 1986, for more detailed discussion of diophantine equations).

Thus, we have above three equivalent statements of our shelf-packing problem. Given the divisibility of shelf length by the items' sizes, when can we pack the shelves ? That is, what happens if we drop just the requirement that $\{H_i \setminus \bar{H}\} \cap \{H_j \setminus \bar{H}\} = \emptyset$, for all i and j, $(i \neq j)$? Given a problem instance, we would like to know if there is a solution to the system (1) - (2). How likely are we to get an affirmative answer without testing for sufficiency conditions ?

If we limit the number of shelves to 2, then given that $\sum_{i=1}^{n} a_i v_i = 2L$, we are asking for the likelihood of a solution to the PARTITION problem given the divisibility property $(v_i | L, \forall i)$. In case there exists a packing failure for $2L$, then given $mL, m > 2$, the packing failure instance for $2L$ can be 'augmented' to a case of mL – incurring a similar packing failure. However, the $mL, m > 2$ case might contain packing failures which do not correspond to an augmentation of the $2L$ packing failure instance. This however does not mean that the likelihood of a packing failure for $mL, m > 2$ increases with m since there might be a larger number of successful packing instances. In this paper we only investigate the $\sum_{i=1}^{n} a_i v_i = 2L$ case for all L such that $1 \leq L \leq 1000$.

The rest of the paper is primarily a description of computational study for a counting procedure which addresses the PARTITION existence question, concluding with a few observations, conjectures, and open problems. We assume throughout this note that for a given L, all the corresponding v_i values are ordered in an increasing order.

3 Counting solutions and failures

Counting shelving solutions with divisibility conditions as above could be very time consuming. In order to so so, we explicitly compute the divisors of L, enumerate the number of different successful shelving solutions, and the number of failures. For instance, given $L = 132$, $2L = 264$. In this case there are 12 divisors: $[1, 2, 3, 4, 6, 11, 12, 22, 33, 44, 66, 132]$, which generate a total of **235,763,724** solutions which, when items' sizes add up to $2L$, also pack $L = 132$. However, in this case there are 8 shelving failures. That is, when divisors multiplied by corresponding frequencies add up to

$2L = 264$, but without a partition – packing solution for each of the two shelves. For instance: $3 + 11 + 8 \times 12 + 2 \times 44 + 66 = 264$, but with no subset that adds up to 132. The likelihood of a shelving failure in this case is 0.0000003.

The first such shelving failure case occurs when $L = 30$. There are 8 divisors and the number of successful shelvings is **14,188**. Still, there is one shelving failure: The participating sizes are 1,6,6,6,6,10,10,15. The total adds up to 60 but does not contain a partition which adds up to 30. The next shelving failure occurs with $L = 60$ — there are 2 such failures. The third shelving failure occurs when $L = 66$ (one failure), and the failure after occurs when $L = 84$ — there are three shelving failures. Still, there are **30,620,402** number of successful shelvings for $L = 84$ when the items' sizes add up to $2L = 168$, which means that the likelihood of a shelving failure is **3/30,620,402** — practically 0.

In order to evaluate the frequency of shelving failures for each given shelf value $L, 1 \leq L \leq 1000$, first we have to count the number of solutions to $\sum_{i=1}^{n_L} v_i x_i = 2L, x_i \in Z^+ \cup 0$, where n_L is the number of divisors of L (easily computable using Euler's formula). One suitable solutions counting methodology known as Dynamic Programming (see Bellman and Dreyfus, 1962, and Denardo, 1982) uses recursive relations as stated below.

The following dynamic programming was implemented to count the number of solutions to $\sum_{i=1}^{n_L} v_i x_i = 2L, x_i \geq 0$ and integer :

Let $f_k(b)$ = the number of different solutions to $\sum_{i=1}^{k} v_i x_i = b, x_i \geq 0$ and integer.

Algorithm count solutions
Input (Positive Integers) : $L,\ n_L,\ v_k, k = 1, \ldots, n_L$.
Initialize counting of solutions by setting $f_k(0) = 1, k = 1, \ldots, n_L$, and $f_0(b) = 0$ for all $b \geq 0$.

for $k \leq n_L, 0 \leq b \leq 2L$

$$\text{Let } f_k(b) = \begin{cases} f_{k-1}(b) & \text{if }, 0 \leq k \leq n_L, 0 \leq b < v_k; \\ f_{k-1}(b) + f_k(b - v_k) & \text{if } v_k \leq b \leq 2L. \end{cases}$$

Output : $f_{n_L}(2L)$

Once the value of $f_{n_L}(2L)$ has been computed, the issue of counting the number of failures boils down to counting how many of these solutions do not contain a PARTITION (i.e, how many do not have a solution to $\sum_{i=1}^{n_L} v_i x_i = L, x_i \geq 0$ and integer).

To the best of our knowledge, there is no dynamic programming formulation which would count (evaluate) the number of failures given an input of L, n_L, and $v_k, k = 1, \ldots, n_L$. Thus, we are left with the task of explicitly enumerating all the PARTITION failures. First, we define what we call the *boundedness condition.*

Definition : For a given input L, n_L, and the divisors $v_i, i = 1, \ldots, n_L$, let $\hat{x}_i$ be the smallest positive integer such that $\hat{x}_i v_i = v_j$, for some $j, i < j \leq n_L$. And $\hat{x}_{n_L} = 1$. The set of positive integers $x_i, i = 1, \ldots, n_L$ satisfies the *boundedness property* if $x_i \leq \hat{x}_i - 1, i = 1, \ldots, n_L$.

Example 6 : Let $L = 30$. The set of divisors of L is $1, 2, 3, 5, 6, 10, 15, 30$. In this case, the boundedness property says that $x_1 \leq 1, x_2 \leq 2, x_3 \leq 1, x_4 \leq 1, x_5 \leq 4, x_6 \leq 2, x_7 \leq 1, x_8 \leq 0$. The shelving failure occurs for the integer vector $\mathbf{x} = (1, 0, 0, 0, 4, 2, 1, 0)$.

Lemma 1 : If there is a PARTITION failure, then there is a failure satisfying the boundedness property.

Proof : Suppose that the vector of integers $\mathbf{x}$ is a failure but packs $2L$. If $x_i \geq \hat{x}_i$, we can reduce x_i by $\hat{x}_i$ and add 1 to x_j for which $\hat{x}_i v_i = v_j$. That is, the new failure would correspond to the integer vector $\mathbf{x}'$ such that, $x_k := x_k$, for $k \neq i, j$, and $x_i := x_i - \hat{x}_i, x_j := x_j + 1$. Clearly if the new vector $\mathbf{x}'$ has a PARTITION solution so does the vector $\mathbf{x}$. Repeating this transformation eventually yields a failure satisfying the boundedness property. □

This lemma allows for faster identification of shelf sizes with PARTITION failures by checking for failures satisfying the boundedness property. Essentially, counting all the failures is done in two stages. In Stage 1, for a given L, we generate all the failures satisfying the boundedness property, and if there are none we increment L and repeat. If the list of failures from Stage 1 is not empty, in Stage 2 we expand the initial list of failures to include all the failures.

Let LIST(L^1) be the list of all failures for a shelf of size L satisfying boundedness conditions. This list can be generated in lexicographical order for solutions to $\sum_{i=1}^{n_L} x_i v_i = 2L, x_i \geq 0$ and integer, satisfying the boundedness conditions, and checking each solution in turn for PARTITION failure.

In Stage 2, we start (in increasing order of L) with the nonempty LIST(L^1) set, and generate a complete set of failures in LIST(L^2) by "lexicographically" expanding each solution in LIST(L^1) and checking each expansion for failure.

Example 7 : $L = 30$, the corresponding v_i's are $(1, 2, 3, 5, 6, 10, 15, 30)$. LIST($(L = 30)^1$) = $\{(1, 0, 0, 0, 4, 2, 1, 0)\}$.

Expansion Routine : Since $x_7 = 1$, create a new solution by setting $x_7 = 0$ and $x_4 := x_4 + 3$. This new solution corresponds to $(1, 0, 0, 3, 4, 2, 0, 0)$ which can be packed into two shelves. No other expansion of $x_7 = 1$ will lead to a packing failure. Next, create a new solution by taking $x_6 := x_6 - 1 = 2 - 1 = 1$ and raising $x_4 := x_4 + 2 = 0 + 2 = 2$. This corresponds to solution $(1, 0, 0, 2, 4, 1, 1, 0)$ which again packs two shelves. Next expand by changing $x_5 := x_5 - 1 = 4 - 1 = 3$, and setting $x_4 := x_4 + 1 = 0 + 1 = 1$ and $x_1 := x_1 + 1 = 1 + 1 = 2$. This corresponds to $(2, 0, 0, 1, 3, 2, 1, 0)$ which packs two shelves. This concludes the search for failures to $L = 30$. Thus, for $L = 30$ we have only one failure – the original member of LIST(L^1).

Example 8 : $L = 60$. Divisors of $L = 60$: $(1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60)$
Boundness conditions for $L = 60$; $(1, 1, 1, 2, 1, 1, 1, 4, 1, 2, 1, 0)$

UPPERBOUND FAILURES – LIST(L^1) ;
$(0, 1, 0, 0, 0, 0, 0, 4, 0, 2, 1, 0)$
$(0, 1, 0, 0, 1, 0, 0, 4, 1, 1, 1, 0)$

ALL FAILURES – LIST (L^2) ;
$(0, 1, 0, 0, 0, 0, 0, 4, 0, 2, 1, 0)$
$(0, 1, 0, 0, 1, 0, 0, 4, 1, 1, 1, 0)$
$(2, 0, 0, 0, 0, 0, 0, 4, 0, 2, 1, 0)$
$(0, 1, 0, 0, 0, 0, 0, 4, 2, 2, 0, 0)$

$(0, 1, 0, 0, 1, 0, 0, 4, 3, 1, 0, 0)$

Tables 1, 2, 3, and 4, present our full evaluation of all the failures when packing with divisible items two shelves of length L each for $1 \leq L \leq 1000$. We list the Ls with failures, the number of divisors for each such L, the total number of solutions satisfying upper bounds, the total number of failures, the total number of packing solutions, and the different times (in cpu seconds) to compute.

In Theorem 2 we identify an interesting case which assures no packing failures. This result inspired two conjectures. We end the paper with a short list of interesting open problems.

Table 1.

L	No. of divisors of L	No. of Solutions $\leq UB$	Total No. of Failures	Time to Compute All Failures	Time to compute Failures $Sat.UB$	Total Time $(E+F)$	Total Solutions Packing $2L$	Failures divided by total Solutions
30	8	5	1	0	0.01	0.01	14188	$7.04821E-05$
60	12	77	5	0.17	0.01	0.18	7173704	$6.9699E-07$
66	8	5	1	0	0	0	117202	$8.53228E-06$
84	12	73	5	0.17	0.03	0.2	30620402	$1.6329E-07$
90	12	105	7	0.261	0.02	0.281	38716091	$1.80803E-07$
102	8	5	1	0.01	0.01	0.02	400268	$2.49833E-06$
105	8	46	3	0.08	0.02	0.1	317755	$9.44124E-06$
120	16	767	22	3.325	0.101	3.426	12293218763	$1.7896E-09$
126	12	101	3	0.34	0.02	0.36	167032015	$1.79606E-08$
130	8	9	1	0.03	0.02	0.05	674172	$1.4833E-06$
132	12	69	22	0.301	0.08	0.381	235763724	$9.33138E-08$
138	8	5	1	0.01	0.01	0.02	954586	$1.04757E-06$
140	12	129	1	0.45	0.02	0.47	249640646	$4.00576E-09$
150	12	87	7	0.311	0.02	0.331	399308813	$1.75303E-08$
154	8	13	3	0.03	0.02	0.05	1031727	$2.90775E-06$
165	8	43	2	0.121	0.01	0.131	1110148	$1.80156E-06$
168	16	729	31	4.676	0.17	4.846	92877707440	$3.33772E-10$
174	8	5	1	0.01	0.01	0.02	1871356	$5.34372E-07$
180	18	5247	94	39.397	0.681	40.078	$1.34E+12$	$7.01022E-11$
182	8	13	5	0.04	0.02	0.06	1652126	$3.0264E-06$
190	8	9	3	0.03	0.02	0.05	1991897	$1.5061E-06$
198	12	97	4	0.481	0.02	0.501	1299047757	$3.07918E-09$
204	12	69	5	0.45	0.03	0.48	1807794705	$2.7658E-09$
210	16	11139	201	87.696	1.432	89.128	$3.40566E+11$	$5.90194E-10$
228	12	69	5	0.431	0.041	0.472	3067593880	$1.62994E-09$
230	8	9	1	0.04	0.01	0.05	3460004	$2.89017E-07$
234	12	93	2	0.571	0.02	0.591	2827127355	$7.07432E-10$
240	20	6687	120	81.367	1.161	82.528	$9.90863E+13$	$1.21107E-12$
246	8	5	1	0.03	0.01	0.04	5157052	$1.93909E-07$
252	18	5047	133	58.695	1.423	60.118	$1.35131E+13$	$9.84229E-12$
255	8	43	1	0.21	0.01	0.22	3834500	$2.6079E-07$
260	12	121	14	0.861	0.06	0.921	4249449808	$3.29454E-09$
264	16	697	83	7.231	0.45	7.681	$1.59954E+12$	$5.189E-11$
270	16	1457	37	15.552	0.221	15.773	$1.63985E+12$	$2.25631E-11$
273	8	64	4	0.321	0.05	0.371	4404675	$9.08126E-07$
276	12	69	22	0.6	0.17	0.77	7656085986	$2.87353E-09$
280	16	1241	43	13.33	0.451	13.781	$1.73739E+12$	$2.47498E-11$
282	8	5	1	0.05	0.02	0.07	7708378	$1.29729E-07$
286	8	21	1	0.12	0.01	0.13	5765522	$1.73445E-07$
290	8	9	3	0.05	0.06	0.11	6793364	$4.41607E-07$
300	18	4719	50	64.082	0.381	64.463	$5.41939E+13$	$9.22614E-13$
306	12	93	32	0.831	0.19	1.021	10024523941	$3.19217E-09$
308	12	173	27	1.472	0.201	1.673	8565643642	$3.15213E-09$
312	16	689	9	8.232	0.23	8.462	$4.7159E+12$	$1.90844E-12$
315	12	1801	20	20.83	0.23	21.06	7003957386	$2.85553E-09$
318	8	5	1	0.06	0.01	0.07	10986956	$9.1017E-08$
322	8	13	1	0.09	0.02	0.11	8500102	$1.17646E-07$
330	16	10903	287	143.115	3.334	146.449	$5.91297E+12$	$4.85374E-11$
336	20	6411	114	113.273	1.382	114.655	$1.32765E+15$	$8.58659E-14$
340	12	117	9	1.382	0.081	1.463	15092391192	$5.96327E-10$
348	12	69	5	0.811	0.05	0.861	23435383392	$2.13353E-10$
350	12	265	7	2.414	0.06	2.474	14680720521	$4.76816E-10$
354	8	5	1	0.06	0.01	0.07	15083986	$6.62955E-08$
360	24	189743	718	5010.95	20.099	5031.049	$2.87494E+17$	$2.49744E-15$

Table 2.

L	No. of divisors of L	No. of Solutions $\leq UB$	Total No. of Failures	Time to Compute All Failures	Time to compute Failures $Sat.UB$	Total Time $(E+F)$	Total Solutions Packing $2L$	Failures divided by total Solutions
364	12	173	28	1.923	0.21	2.133	18697287678	$1.49754E-09$
372	12	69	5	0.961	0.07	1.031	32385184878	$1.54392E-10$
374	8	21	5	0.16	0.061	0.221	12407155	$4.02993E-07$
378	16	1393	35	20.309	0.29	20.599	$1.26951E+13$	$2.75696E-12$
380	12	117	65	1.512	0.421	1.933	25656216568	$2.5335E-09$
385	8	366	1	3.315	0.02	3.335	10336862	$9.67412E-08$
390	16	10827	217	185.957	2.724	188.681	$1.74721E+13$	$1.24198E-11$
396	18	4845	192	93.114	2.193	95.307	$3.4872E+14$	$5.50585E-13$
399	8	61	5	0.481	0.04	0.521	13095705	$3.81805E-07$
408	16	681	41	13.74	0.551	14.291	$2.74666E+13$	$1.49272E-12$
414	12	93	7	1.682	0.06	1.742	42585943859	$1.64373E-10$
420	24	1471419	4211	47665.8	306.1	47971.9	$1.30103E+18$	$3.23668E-15$
426	8	5	1	0.08	0.03	0.11	26098202	$3.83168E-08$
429	8	100	3	0.931	0.05	0.981	15509999	$1.93424E-07$
430	8	9	1	0.11	0.03	0.14	21576852	$4.6346E-08$
435	8	43	9	0.431	0.111	0.542	18084826	$4.97655E-07$
440	16	1185	13	22.362	0.45	22.812	$3.05665E+13$	$4.25302E-13$
442	8	25	3	0.251	0.03	0.281	19898386	$1.50766E-07$
450	18	9909	92	266.584	0.902	267.486	$8.09397E+14$	$1.13665E-13$
455	8	366	6	4.566	0.11	4.676	16629274	$3.60809E-07$
456	16	673	43	16.865	0.491	17.356	$5.74816E+13$	$7.48065E-13$
460	12	117	5	1.792	0.07	1.862	64205222718	$7.78753E-11$
462	16	15159	248	384.944	3.565	388.509	$4.59448E+13$	$5.39778E-12$
468	18	4803	86	110.238	2.073	112.311	$1.19883E+15$	$7.17365E-14$
480	24	55643	496	2399.87	12.358	2412.228	$4.72098E+18$	$1.05063E-16$
490	12	241	6	4.697	0.06	4.757	74056883152	$8.10188E-11$
492	12	69	5	1.652	0.08	1.732	$1.26482E+11$	$3.95312E-11$
495	12	1720	48	29.552	0.631	30.183	56162238952	$8.54667E-10$
498	8	5	1	0.16	0.03	0.19	41480626	$2.41076E-08$
504	24	183969	1073	7760.95	35.631	7796.581	$6.78142E+18$	$1.58226E-16$
510	16	10739	254	267.404	4.697	272.101	$1.0206E+14$	$2.48873E-12$
516	12	69	5	1.452	0.13	1.582	$1.59669E+11$	$3.13148E-11$
518	8	13	1	0.17	0.02	0.19	34109845	$2.9317E-08$
520	16	1169	146	24.305	1.862	26.167	$9.06521E+13$	$1.61055E-12$
522	12	93	4	1.702	0.05	1.752	$1.30595E+11$	$3.0629E-11$
525	12	2728	33	39.177	0.591	39.768	75010782794	$4.39937E-10$
528	20	6143	354	174.03	6.449	180.479	$5.11681E+16$	$6.91837E-15$
530	8	9	1	0.14	0.03	0.17	39991688	$2.50052E-08$
532	12	169	1	2.744	0.05	2.794	$1.13233E+11$	$8.83133E-12$
534	8	5	1	0.12	0.03	0.15	51037916	$1.95933E-08$
540	24	272213	946	10248.7	32.106	10280.806	$1.49555E+19$	$6.32543E-17$
546	16	15035	350	354.48	5.989	360.469	$1.3611E+14$	$2.57144E-12$
550	12	253	9	3.815	0.12	3.935	$1.16513E+11$	$7.72446E-11$
552	16	665	296	16.784	4.306	21.09	$2.06264E+14$	$1.43505E-12$
555	8	43	3	0.57	0.05	0.62	36969879	$8.11471E-08$
558	12	93	2	1.953	0.04	1.993	$1.8055E+11$	$1.10772E-11$
560	20	10775	232	324.086	4.367	328.453	$5.76448E+16$	$4.02465E-15$
564	12	69	22	1.672	0.39	2.062	$2.46869E+11$	$8.91159E-11$
570	16	10691	278	260.805	4.857	265.662	$2.13805E+14$	$1.30025E-12$
572	12	273	8	4.847	0.151	4.998	$1.48113E+11$	$5.40129E-11$
574	8	13	5	0.201	0.07	0.271	46137385	$1.08372E-07$
580	12	117	20	2.303	0.19	2.493	$1.97057E+11$	$1.01493E-10$
585	12	1702	4	35.641	0.15	35.791	$1.23266E+11$	$3.24502E-11$
588	18	4225	62	122.977	1.162	124.139	$7.21195E+15$	$8.59685E-15$

Table 3.

L	No. of divisors of L	No. of Solutions $\leq UB$	Total No. of Failures	Time to Compute All Failures	Time to compute Failures $Sat.UB$	Total Time $(E+F)$	Total Solutions Packing $2L$	Failures divided by total Solutions
590	8	9	3	0.17	0.05	0.22	54922829	$5.46221E-08$
594	16	1329	28	30.524	0.561	31.085	$2.23668E+14$	$1.25185E-13$
595	8	356	14	4.306	0.23	4.536	35958538	$3.89337E-07$
598	8	25	3	0.361	0.09	0.451	47696988	$6.28971E-08$
600	24	176269	519	7921.18	15.713	7936.893	$8.49976E+18$	$6.10606E-17$
606	8	5	1	0.14	0.04	0.18	74340652	$1.34516E-08$
612	18	4733	303	163.655	8.773	172.428	$8.96934E+15$	$3.37818E-14$
615	8	43	2	0.681	0.05	0.731	50021278	$3.9983E-08$
616	16	1673	208	42.762	5.007	47.769	$2.40816E+14$	$8.6373E-13$
624	20	6065	106	209.561	3.165	212.726	$2.04914E+17$	$5.1729E-16$
630	24	5037267	5906	273892	634.943	274526.943	$1.50433E+19$	$3.926E-16$
636	12	69	5	4.115	0.151	4.266	$4.45248E+11$	$1.12297E-11$
638	8	21	3	0.351	0.22	0.571	58451001	$5.1325E-08$
642	8	5	1	0.19	0.04	0.23	88268498	$1.13291E-08$
644	12	165	25	5.068	0.661	5.729	$2.837E+11$	$8.81212E-11$
650	12	249	34	5.909	0.491	6.4	$2.54984E+11$	$1.33342E-10$
651	8	61	5	0.951	0.1	1.051	54566499	$9.16313E-08$
660	24	1476755	6790	83675.9	613.392	84289.292	$1.7818E+19$	$3.81075E-16$
665	8	356	2	5.999	0.05	6.049	49623939	$4.03031E-08$
672	24	53445	431	3124.69	12.979	3137.669	$1.87766E+18$	$2.29541E-16$
678	8	5	1	0.2	0.04	0.24	103835596	$9.63061E-09$
680	16	1153	90	36.082	1.412	37.494	$5.32083E+14$	$1.69147E-13$
684	18	4691	60	181.531	2.173	183.704	$2.08493E+16$	$2.8778E-15$
690	16	10619	347	381.539	8.071	389.61	$7.68362E+14$	$4.5161E-13$
693	12	2431	17	55.439	0.381	55.82	$2.53398E+11$	$6.70882E-11$
696	16	665	22	22.593	0.651	23.244	$9.84946E+14$	$2.23363E-14$
700	18	13015	97	441.795	2.073	443.868	$1.61871E+16$	$5.99243E-15$
702	16	1305	31	36.172	0.531	36.703	$6.63565E+14$	$4.67173E-14$
705	8	43	1	0.831	0.02	0.851	74853400	$1.33594E-08$
708	12	69	22	2.423	0.581	3.004	$7.54551E+11$	$2.91564E-11$
714	16	14903	169	482.614	4.766	487.38	$7.97732E+14$	$2.11851E-13$
720	30	6349305	6177	826.859	4242.56	5069.419	$1.35579E+19$	$4.55601E-16$
726	12	71	13	1.673	0.141	1.814	$6.25275E+11$	$2.07909E-11$
728	16	1657	155	57.592	3.835	61.427	$7.16024E+14$	$2.16473E-13$
730	8	9	1	0.27	0.04	0.31	103243734	$9.68582E-09$
735	12	2545	34	58.584	0.771	59.355	$3.80423E+11$	$8.93741E-11$
738	12	93	7	3.105	0.141	3.246	$7.0629E+11$	$9.91094E-12$
740	12	117	9	3.795	0.26	4.055	$6.45635E+11$	$1.39398E-11$
741	8	118	13	2.274	0.3	2.574	74265156	$1.75048E-07$
742	8	13	3	0.36	0.06	0.42	98428999	$3.04788E-08$
744	16	665	31	27.92	1.032	28.952	$1.54798E+15$	$2.0026E-14$
748	12	269	28	7.431	0.631	8.062	$5.28355E+11$	$5.29947E-11$
750	16	1131	103	33.588	1.382	34.97	$1.34706E+15$	$7.64626E-14$
756	24	262753	1495	16404.4	67.968	16472.368	$7.62241E+18$	$1.96132E-16$
759	8	97	4	1.803	0.11	1.913	80470054	$4.97079E-08$
760	16	1145	249	42.571	4.276	46.847	$1.11652E+15$	$2.23014E-13$
765	12	1675	15	44.033	0.481	44.514	$4.41967E+11$	$3.39392E-11$
770	16	72107	348	2837.11	10.896	2848.006	$9.14894E+14$	$3.80372E-13$
774	12	93	3	3.425	0.05	3.475	$8.91815E+11$	$3.36393E-12$
780	24	1476267	4914	331.457	9207.14	9538.597	$5.0368E+18$	$9.756194E-16$
786	8	5	1	0.28	0.05	0.33	161284402	$6.20023E-09$
790	8	9	3	0.291	0.08	0.371	130532459	$2.29828E-08$
792	24	177985	1849	11546.5	98.992	11645.492	$8.36475E+18$	$2.21047E-16$
798	16	14831	264	579.143	6.579	585.722	$1.6731E+15$	$1.57791E-13$
804	12	69	5	3.115	0.261	3.376	$1.41187E+12$	$3.54141E-12$

Table 4.

L	No. of divisors of L	No. of Solutions $\leq UB$	Total No. of Failures	Time to Compute All Failures	Time to compute Failures $Sat.UB$	Total Time $(E+F)$	Total Solutions Packing $2L$	Failures divided by total Solutions
805	8	351	8	6.349	0.19	6.539	86526866	$9.24568E-08$
806	8	25	1	0.531	0.03	0.561	114012117	$8.771E-09$
810	20	18273	294	859.516	4.236	863.752	$1.6346E+18$	$1.7986E-16$
816	20	5969	189	318.859	4.897	323.756	$1.96684E+18$	$9.60934E-17$
819	12	2404	60	68.168	2.884	71.052	$5.57088E+11$	$1.07703E-10$
822	8	5	1	0.351	0.06	0.411	184320508	$5.42533E-09$
825	12	2602	33	72.825	1.342	74.167	$6.0496E+11$	$5.45491E-11$
828	18	4677	220	222.24	7.081	229.321	$8.97268E+16$	$2.45189E-15$
830	8	9	1	0.421	0.04	0.461	151164764	$6.6153E-09$
836	12	269	1	8.833	0.07	8.903	$8.99533E+11$	$1.11169E-12$
840	32	182206877	75829	100.432	2238.50	2338.9332	$1.57659E+19$	$4.80968E-15$
846	12	93	4	4.316	0.09	4.406	$1.37943E+12$	$2.89975E-12$
852	12	69	22	4.206	0.822	5.028	$1.8795E+12$	$1.17052E-11$
855	12	1666	30	55.58	0.841	56.421	$7.53789E+11$	$3.97989E-11$
858	16	23459	715	1009.86	23.714	1033.574	$2.41185E+15$	$2.96453E-13$
860	12	117	5	4.727	0.16	4.887	$1.34709E+12$	$3.71169E-12$
861	8	61	18	1.713	0.421	2.134	124214728	$1.4491E-07$
868	12	165	5	5.818	0.26	6.078	$1.20579E+12$	$4.14667E-12$
870	16	10571	2093	478.698	63.241	541.939	$3.6745E+15$	$5.69601E-13$
880	20	10317	239	536.862	8.922	545.784	$2.28712E+18$	$1.04498E-16$
882	18	8841	40	373.577	1.402	374.979	$1.1031E+17$	$3.62614E-16$
884	12	321	43	11.136	0.792	11.928	$1.15634E+12$	$3.71863E-11$
885	8	43	9	1.602	0.26	1.862	146708332	$6.13462E-08$
888	16	665	9	34.42	0.831	35.251	$5.15896E+15$	$1.74454E-15$
890	8	9	3	0.45	0.1	0.55	186020000	$1.61273E-08$
894	8	5	1	0.4	0.06	0.46	236769676	$4.22351E-09$
900	27	3457471	3339	261566	7748.59	8010.156	$1.3695E+19$	$2.43811E-16$
910	16	71659	847	3387.82	32.196	3420.016	$2.72628E+15$	$3.1068E-13$
912	20	5937	247	363.974	7.921	371.895	$5.0758E+18$	$4.86623E-17$
918	16	1289	152	57.813	2.754	60.567	$3.89638E+15$	$3.90105E-14$
920	16	1137	45	55.3	2.234	57.534	$4.02246E+15$	$1.11872E-14$
924	24	2068531	8786	889.921	12062.5	12960.421	$1.40016E+19$	$6.27499E-15$
930	16	10559	225	547.818	5.628	553.446	$5.77714E+15$	$3.89466E-14$
936	24	176307	908	14802.1	49.951	14852.051	$1.26354E+19$	$7.18615E-17$
938	8	13	3	0.611	0.1	0.711	197083249	$1.5222E-08$
940	12	117	1	6.099	0.07	6.169	$2.08402E+12$	$4.79841E-13$
945	16	53367	141	2532.55	6.209	2538.759	$2.4818E+15$	$5.68135E-14$
946	8	21	9	0.721	0.23	0.951	185953630	$4.83992E-08$
948	12	69	5	4.857	0.341	5.198	$3.18461E+12$	$1.57005E-12$
950	12	245	22	8.442	0.34	8.782	$1.55122E+12$	$1.41824E-11$
952	16	1625	4	84.382	0.18	84.562	$4.21675E+15$	$9.48598E-16$
954	12	93	32	6.329	1.032	7.361	$2.48914E+12$	$1.28558E-11$
960	28	453389	1918	89.579	1337.39	1426.969	$1.10661E+19$	$1.73322E-16$
962	8	25	1	1.022	0.04	1.062	191680631	$5.21701E-09$
966	16	14771	191	874.147	9.654	883.801	$6.02301E+15$	$3.17117E-14$
969	8	154	3	5.087	0.1	5.187	160481808	$1.86937E-08$
975	12	2566	18	93.184	0.721	93.905	$1.32986E+12$	$1.35353E-11$
980	18	12253	95	658.938	3.115	662.053	$2.13139E+17$	$4.45718E-16$
984	16	665	41	45.936	1.902	47.838	$1.03993E+16$	$3.94257E-15$
986	8	33	1	1.241	0.04	1.281	202100472	$4.94803E-09$
987	8	61	3	2.433	0.131	2.564	185920511	$1.61359E-08$
988	12	321	3	14.741	0.25	14.991	$1.96908E+12$	$1.52356E-12$
990	24	5029899	10136	1262.08	2782.76	4044.84	$5.98483E+18$	$1.69361E-14$
996	12	69	22	5.998	1.372	7.37	$4.06536E+12$	$5.41158E-12$

Theorem 2 : Suppose that $L = p^r q$ for some integer $r > 0$ and for prime numbers p and q. Then

L has no failures.

Proof : Let $v_{ij} = p^i q^j$ for $i = 0, 1, \dots, r$ and $j = 0, 1$. These are all of the factors of L. Let x_{ij} be the number of times that v_{ij} appears in the partition of $2L$. If L has a failure, then it has a failure x satisfying the bounds below:

$x_{ij} \leq (p-1)$ for $i = 0, \dots, r-1$ and $j = 0, 1$.
$x_{r0} \leq q - 1$
$x_{r1} = 0$.

But then $\sum_{i=0}^{r} \sum_{j=0}^{1} v_{ij} x_{ij} \leq (p-1)(1 + \cdots + p^{r-1})(1+q) + p^r(q-1)$
$= (p^r - 1)(1+q) + p^r(q-1) = 2p^r q - q - 1 \leq 2L$. □

Corollary 1 : If L has a failure, then L has at least 8 divisors.

Proof : If L has three distinct prime factors it has at least 8 divisors. If it has two prime factors, they both must have multiplicity at least 2, and so there are at least 9 divisors. □

Conjecture 1 : If L has only two prime divisors, then there are no failures.

Conjecture 2 : There is a positive integer K (finite) such that every positive integer with at least K prime factors has a failure.

4 Open Problems

Problem 1 : What is the complexity of the two shelf shelving problem with the divisibility property as above ? For a given integer $L > 0$, is it possible to evaluate solutions with the boundedness property in polynomial time ? Can the evaluation of the expansions of the failures with boundedness property be done in polynomial time ?

Problem 2 : As $L \to \infty$ does the ratio of failures to shelving solutions, over the set of Ls with failures, go to zero ?

References

Bellman, R. and S. Dreyfus, (1962). *Applied Dynamic Programming*, Princeton, NJ: Princeton University Press.

Denardo, E., (1982). *Dynamic Programming: Theory and Applications*, Englewood Cliffs, NJ: Prentice Hall.

Dror, M., B. Smith, and M. Trudeau, (1994). "Utilizing Shelve Slots: Sufficiency Conditions for Some Easy Instances of Hard Problems", *Journal of Complexity* 10, 216-229.

Garey, M.R. and D.S. Johnson, (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman: San Francisco, CA.

Schrijver, A., (1986). *Theory of Linear and Integer Programming*, Wiley, New York.