



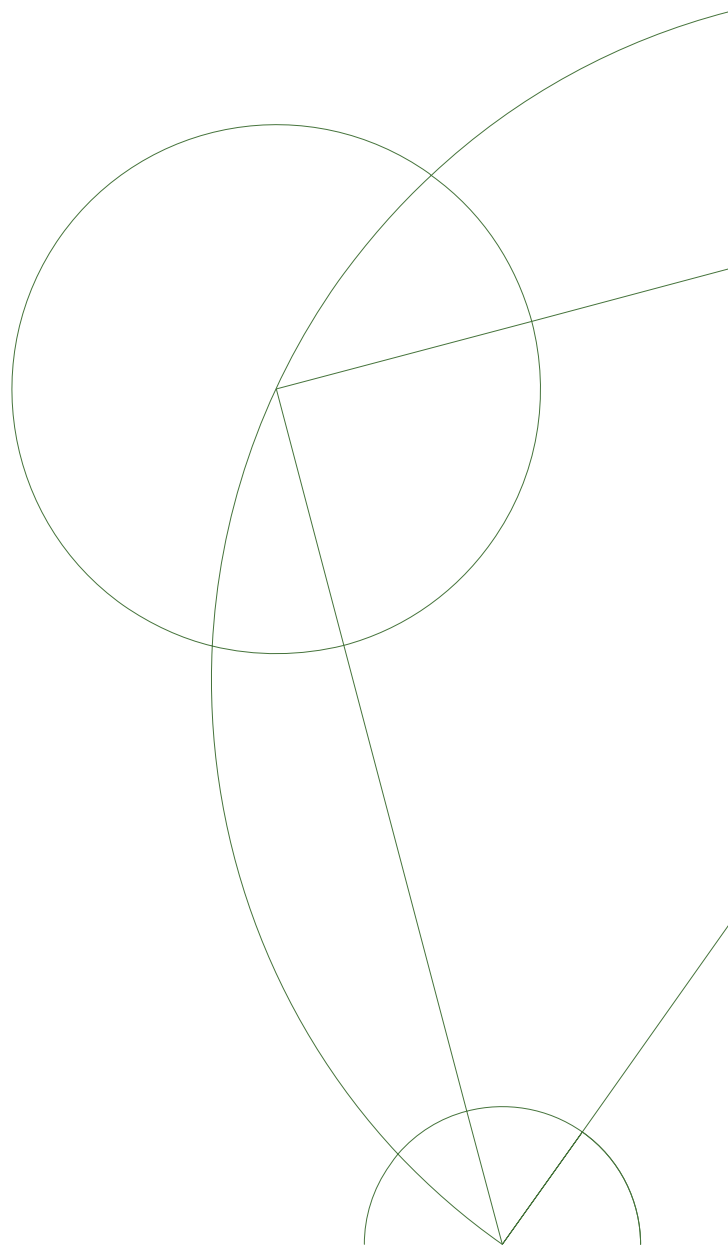
An Experimental Approach to Packing Problems

Nikolaj Ingemann von Holck

Bachelor thesis

Advisor: Søren Eilers

January 12, 2018



ABSTRACT

This thesis employs experimental mathematics to examine a multidimensional packing problem proposed by Dean G. Hoffman and based on the AM-GM-inequality. We formalize the problem and generalize several of Hoffman's results from the three-dimensional case to higher dimensions. We explain why it is complicated to count the number of unique packings and introduce the notion of a universal packing to remedy this. In the end, this enables us to provide the first rigorous proof that the three-dimensional case can always be solved.

We reproduce most of the results presented in the sparse literature on the subject, provide a rectified count of the number of unique squares in the four-dimensional case and give an estimate of the computations needed to determine the number of cubes in the four-dimensional case. We also construct a four-dimensional universal packing, formulate several hypotheses closely related to Maclaurin's inequality and prove one of these to be a necessary condition for being a universal packing. In addition, we use this universal packing as a stepping stone to give a lower bound on the number of unique four-dimensional universal packings.

Finally, we discuss how to approach the five-dimensional case. This includes ways of exploiting the hypotheses formulated during the investigation of the four-dimensional case and an assessment of the viability of the methods utilized to compute packings in lower dimensions.

CONTENTS

1. Introduction	2
2. Problem formulation and analysis	5
2.1. Equivalence of packings	7
2.2. Universal packings	9
3. Properties of universal packings	14
3.1. Verifying a Hoffman packing	16
3.2. Promoting a Hoffman packing to a universal packing	20
4. Experiments in the three-dimensional case	24
4.1. Dynamic programming	24
4.2. Backtracking	25
4.3. Results and analysis of solutions	26
4.4. Comparison of the two approaches	27
4.5. Producing a packing of any dimension tuple using universal packing	28
5. Experiments in the four-dimensional case	31
5.1. Utilizing approaches from the three-dimensional case	32
5.2. Counting the number of unique squares	34
5.3. Combining two-dimensional packings	37
5.4. The subgrid criterion	37
6. Approaching the five-dimensional case	42
6.1. Fundamentally different approaches	43
6.2. Ties to Maclaurin's inequality	43
6.3. Formulation as a constraint satisfaction problem	44
Appendix A. Three-dimensional universal packings	46
Appendix B. Four-dimensional universal packing	47
References	53
Index	54

1. INTRODUCTION

In [Hof81] Dean G. Hoffman proposes a packing problem based on the inequality of the arithmetic mean and geometric mean. The following introduction is immensely inspired by his presentation of it.

Is it possible to pack four 7-by-8 rectangles inside a square whose sides have a length of 15? Notice that each rectangle has an area of $7 \cdot 8 = 56$, while the area of the square is $15^2 = 225$. Thus, the leftover area is just $225 - 4 \cdot 56 = 1$, so there is a very limited room for maneuver. Nevertheless, it is in fact possible to pack the rectangles inside the square by organizing them like in Figure 1.

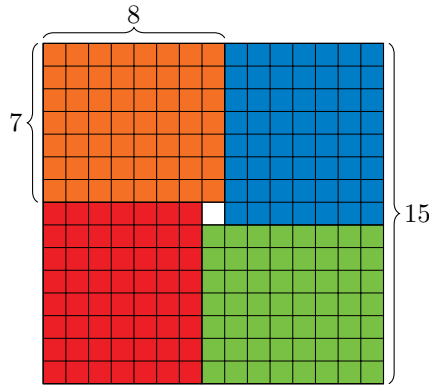


FIGURE 1. Packing of four 7-by-8 rectangles inside a 15-by-15 square.

We can generalize this problem. Let x and y be two positive real numbers. Is it possible to pack four x -by- y rectangles inside a square whose sides have a length of $x + y$? The area of each rectangle is xy , while the area of the square is $(x + y)^2$. We will certainly not be able to fit the rectangles if their combined area exceeds the area of the square. Thus, we do not stand a chance unless $4xy \leq (x + y)^2$. Fortunately, there is still a chance of finding a packing, since $0 \leq (x - y)^2 = (x + y)^2 - 4xy$. Inspired by Figure 1, we can solve this generalized problem as well. Figure 2 shows how to pack the four rectangles inside the square, depending of the relative sizes of x and y .

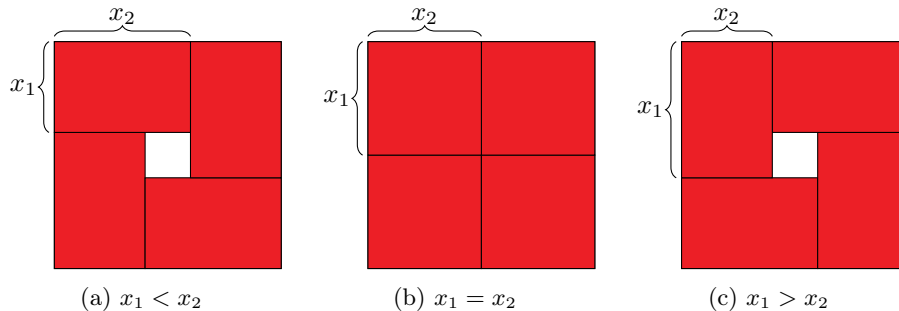


FIGURE 2. Solution to the two-dimensional packing problem.

There is an elegant way to generalize this packing problem to higher dimensions. It is based on the AM-GM inequality, which states that for any n non-negative real numbers x_1, x_2, \dots, x_n , then

$$(1) \quad \sqrt[n]{x_1 \cdot x_2 \cdots x_n} \leq \frac{x_1 + x_2 + \cdots + x_n}{n}$$

with equality if and only if $x_1 = x_2 = \dots = x_n$. The left-hand side is called the *geometric mean* while the right-hand side is the familiar *arithmetic mean*. We can turn this inequality into a packing problem by multiplying both sides by n , and then raising both sides to the power of n . This yields

$$n^n (x_1 \cdot x_2 \cdots x_n) \leq (x_1 + x_2 + \cdots + x_n)^n.$$

Observe that $x_1 \cdot x_2 \cdots x_n$ is the hypervolume of an n -dimensional hyperrectangle with dimensions $x_1 \times x_2 \times \cdots \times x_n$ and that $(x_1 + x_2 + \cdots + x_n)^n$ is the hypervolume of an n -dimensional hypercube whose sides have length $x_1 + x_2 + \cdots + x_n$. So, is it always possible to pack n^n such hyperrectangles inside the n -dimensional hypercube? We say that a dimension is *good* if the answer to this question is yes. The above inequality does not guarantee that the hyperrectangles will fit, but only that their combined hypervolume will not exceed the hypervolume of the hypercube.

We have just seen that 2 is a good dimension and in fact so is 3. In the three-dimensional case there are $3^3 = 27$ bricks and Figure 3 gives a general recipe for constructing a packing. There are in fact 21 such recipes if we ignore reflections and/or rotations.

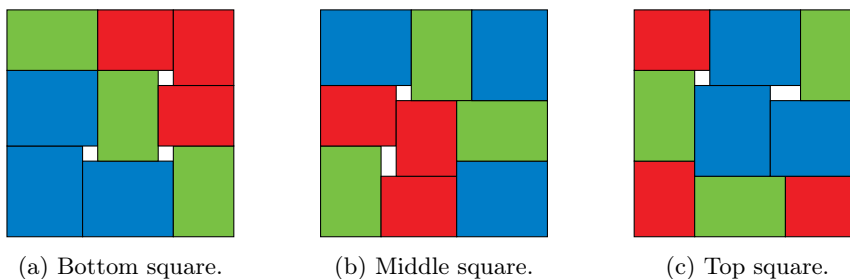


FIGURE 3. Solution to the three-dimensional packing problem. Let (x_1, x_2, x_3) be a dimension tuple with $x_1 \leq x_2 \leq x_3$. Then the x_1 -by- x_2 , x_1 -by- x_3 and x_2 -by- x_3 faces are colored red, green and blue, respectively.

What about higher dimensions? This packing problem has the extraordinary property that if m and n are good dimensions, then mn is good as well. Hence, 4 is a good dimension. It is unknown whether 5 is a good dimension.

The aim of this thesis is to cast light on some of the most interesting unanswered questions of Hoffman's multidimensional packing problem, namely how many packings there are in the four-dimensional case and whether 5 is a good dimension. With the ambition of increasing our knowledge of this problem, we will use experimental mathematics to investigate it.

Experimental mathematics as described in [EJ17, p. 5-6] is inspired by the methodology of the fields of science where experiments play an essential role in obtaining knowledge. While no amount of experimental data can prove a hypothesis as a mathematical truth (unless it has finitely many cases which can be checked one at a time), it can still play a significant role in guiding the research process.

An experimental approach might provide valuable insights into a mathematical problem—insights which can help us put forward conjectures, find counterexamples and gain an understanding, which may ultimately help us formulate a proof. There is a close interplay with theoretical investigations since, in this way, we can examine the implications of our findings, refine our hypotheses and subsequently conduct more experiments. These experiments could, in theory, be carried out by hand. However, the rapidly growing capabilities of modern computers enable us to perform much more complicated and computationally demanding experiments, which

might not previously have been practically feasible to carry out. This continuous expansion of the practical possibilities of experimental mathematics only makes the approach more appealing and promising.

Thanks to Dean G. Hoffman for providing important insights into the inner workings of this packing problem and thanks to Trine K. Boomsma for helping approach the problem using mixed-integer programming and constraint satisfaction.

The repository containing most of the code developed during this project is available at <https://github.com/nikolajholck/hoffman>.

2. PROBLEM FORMULATION AND ANALYSIS

We would like to formalize the formulation of Hoffman’s multidimensional packing problem. Let n be a positive integer denoting the dimension of the problem. We will refer to an element d of $(0, \infty)^n$ as an n -dimensional *dimension tuple* and we refer to the sum of its coordinates as the sum of the dimension tuple. Note that we will typically omit the “ n -dimensional” part of notions, whenever it is clear from the context and for instance simply write “dimension tuple”.

Definition 2.1 (Hyperrectangle). A subset R of \mathbb{R}^n is an n -dimensional *hyperrectangle*, if it can be written as a Cartesian product of n non-empty, open and bounded intervals I_1, I_2, \dots, I_n of \mathbb{R} , that is $R = I_1 \times I_2 \times \dots \times I_n$.

In addition, for each $i = 1, 2, \dots, n$ let a_i and b_i denote the left and right endpoints of the interval $I_i = (a_i, b_i)$, let $x_i = b_i - a_i$ denote the interval width and let $m_i = (a_i + b_i)/2$ denote the interval midpoint. We will refer to the point (a_1, a_2, \dots, a_n) as the *anchor point* and the point (m_1, m_2, \dots, m_n) as the *centre* of the hyperrectangle R . We will refer to the tuple (x_1, x_2, \dots, x_n) as the dimension tuple of the hyperrectangle R and refer to its i ’th element as the *extent* of the hyperrectangle along the i ’th dimension. Lastly, the n -dimensional *hypervolume* of the hyperrectangle R is the product $x_1 x_2 \dots x_n$.

Notice that the hyperrectangle R is uniquely identified by its anchor point and dimension tuple, since this information determines the endpoints of I_i for each $i = 1, 2, \dots, n$. When we are not interested in the “position” of a hyperrectangle in \mathbb{R}^n , we will typically just refer to it by its dimension tuple.

Definition 2.2 (Hypercube). An n -dimensional *hypercube* is an n -dimensional hyperrectangle with the same extent along all dimensions, i.e. $x_1 = x_2 = \dots = x_n$.

Definition 2.3 (General hyperrectangle). A subset of \mathbb{R}^n is an n -dimensional *general hyperrectangle* if it can be rotated into an n -dimensional hyperrectangle.

Two general hyperrectangles are said to be *overlapping* if they intersect one another and *non-overlapping* if they are disjoint.

Definition 2.4 (Packing). Let d be an n -dimensional dimension tuple and let s denote its sum. Suppose C is an n -dimensional hypercube with extent s , and suppose B is a set of n -dimensional general hyperrectangles of \mathbb{R}^n , all of which can be rotated into an n -dimensional hyperrectangle with dimension tuple d . Then (B, C) is a *packing* of d , if

- (i) The general hyperrectangles in B are pairwise non-overlapping.
- (ii) Each general hyperrectangle in B is contained in C .

We will refer to C as the *surrounding* hypercube. Note that we can always translate the general hyperrectangles in B and the surrounding hypercube C of a packing (B, C) , so that the surrounding hypercube has its centre at the origin of the coordinate system. From now on we will assume that such a translation has been performed, unless otherwise stated. We are now ready to formulate the n -dimensional packing problem.

Definition 2.5 (Good dimension). A positive integer n is a good dimension, if for any choice of n -dimensional dimension tuple d there exists a packing of d .

Hoffman asks the following question in [Hof81, p. 222], which according to [BCG04, p. 914] and recent correspondences with Dean G. Hoffman [Hof17] does not appear to have been answered for all dimensions.

Question 2.6. For which positive integers n , is n a good dimension?

Figure 2 demonstrates that 2 is a good dimension by showing how to construct a packing of any two-dimensional dimension tuple. According to [Hof81, p. 215] 3 is good, but Hoffman does not provide a thorough proof. We provide the first thorough proof of this in Corollary 4.7. Theorem 2.7 below shows that this packing problem exhibits an extraordinary property, which in particular implies that the number of good dimensions is unbounded. The first appearance of this result is unclear, but Hoffman attributes it to Raphael Robinson in [Hof81, p. 223], while Elwyn R. Berlekamp et al. [BCG04, p. 914] attribute it to David Seal as well. The proof is based on the one presented by Hoffman in [Hof81, p. 223–225].

Theorem 2.7. Suppose m and n are positive integers. If both m and n are good dimensions, then mn is a good dimension as well.

Proof. Suppose m and n are good dimensions and let

$$(x_{1,1}, x_{1,2}, \dots, x_{1,n}, x_{2,1}, x_{2,2}, \dots, x_{2,n}, \dots, x_{m,1}, x_{m,2}, \dots, x_{m,n})$$

be an mn -dimensional dimension tuple and let t denote its sum. We would like to show that there exists a packing of this dimension tuple. Suppose we have $(mn)^{mn}$ mn -dimensional hyperrectangles with the above dimension tuple. We would like to rotate and translate them to fit inside an mn -dimensional hypercube with an extent of t . For the sake of readability we will write out the dimension tuple in an $m \times n$ matrix

$$A = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}.$$

Let s_i be the sum of the i 'th column of A for $i = 1, 2, \dots, n$. First, divide the $(mn)^{mn}$ mn -dimensional hyperrectangles into groups, each consisting of m^m hyperrectangles. There must be $(mn)^{mn}/m^m = m^{m(n-1)}n^{mn}$ of these groups. Consider the dimension tuple $(x_{1,1}, x_{2,1}, \dots, x_{m,1})$ consisting of the first column of A and recall that its sum is s_1 . Since m is a good dimension, there exists a packing of m^m m -dimensional general hyperrectangles (which can be rotated to have this dimension tuple) inside an m -dimensional hypercube with an extent of s_1 . Then we see that the m^m mn -dimensional hyperrectangles in each group will fit inside an mn -dimensional hyperrectangle with dimension tuple (as a matrix)

$$\begin{pmatrix} s_1 & x_{1,2} & \cdots & x_{1,n} \\ s_1 & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ s_1 & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}.$$

By “fit” we mean that the m^m hyperrectangles are pairwise non-overlapping and rotated and translated such that they are subsets of the surrounding hyperrectangle like in Definition 2.4. Doing this for each group results in $m^{m(n-1)}n^{mn}$ such mn -dimensional hyperrectangles—each containing m^m of the original mn -dimensional hyperrectangles, rotated and translated appropriately. We repeat this procedure for every column, each time dividing the mn -dimensional hyperrectangles constructed at the previous iteration into groups with a size of m^m and then fitting the hyperrectangles of each group into a surrounding mn -dimensional hyperrectangle. After doing this a total number of n times, we end up with n^{mn} groups of mn -dimensional hyperrectangles each of which fits inside an mn -dimensional hyperrectangle with

dimension tuple (as a matrix)

$$B = \begin{pmatrix} s_1 & s_2 & \cdots & s_n \\ s_1 & s_2 & \cdots & s_n \\ \vdots & \vdots & \ddots & \vdots \\ s_1 & s_2 & \cdots & s_n \end{pmatrix}.$$

Each of these mn -dimensional hyperrectangles contains m^{mn} of the original mn -dimensional hyperrectangles, rotated and translated appropriately. Next, we perform a similar procedure for each row. We divide the n^{mn} mn -dimensional hyperrectangles into groups of n^n hyperrectangles. There are $n^{mn}/n^n = n^{(m-1)n}$ of these groups. Consider the dimension tuple (s_1, s_2, \dots, s_n) consisting of the first row of B and notice that the sum along this row is t . Since n is a good dimension, there exists a packing of n^n n -dimensional general hyperrectangles (which can be rotated to have this dimension tuple) inside an n -dimensional hypercube with an extent of t . Then the n^n mn -dimensional hyperrectangles in each group will fit inside an mn -dimensional hyperrectangle with dimension tuple (as a matrix)

$$\begin{pmatrix} t & t & \cdots & t \\ s_1 & s_2 & \cdots & s_n \\ \vdots & \vdots & \ddots & \vdots \\ s_1 & s_2 & \cdots & s_n \end{pmatrix}.$$

Doing so for each group, we obtain $n^{(m-1)n}$ such mn -dimensional hyperrectangles—each containing $m^{mn}n^n$ of the original mn -dimensional hyperrectangles, rotated and translated appropriately. Repeating this procedure for each row, m times in total, we end up with $n^{(m-m)n} = 1$ group of n^n mn -dimensional hyperrectangles fitting inside an mn -dimensional hyperrectangle with dimension tuple (as a matrix)

$$\begin{pmatrix} t & t & \cdots & t \\ t & t & \cdots & t \\ \vdots & \vdots & \ddots & \vdots \\ t & t & \cdots & t \end{pmatrix}.$$

This is in fact an mn -dimensional hypercube with an extent of t and the desired packing is obtained by repeatedly unwrapping the groups of rotated and translated mn -dimensional hyperrectangles, until we reach the now appropriately rotated and translated $(mn)^{mn}$ mn -dimensional hyperrectangles. ■

It follows from this result that in order to prove that all dimensions are good, one would only have to prove so for all prime numbers. Hence, 4 is a good dimension, and in the light of Corollary 4.7 the smallest unsettled dimension is 5 [Hof81, p. 223][BCG04, p. 914].

If n is a good dimension, it is natural to ask how many different packings there are in that particular dimension. Before we can even attempt to answer this question, we need to make sure that the number of different packings is independent of the choice of dimension tuple. This will ultimately lead to the definition of a universal packing and a refinement of the question. In preparation for this, we will first need a notion of two packings being notably different.

2.1. Equivalence of packings. Depending on the choice of dimension tuple d there might be enough unoccupied “room” inside the surrounding hypercube to rearrange the general hyperrectangles without intersecting one another and without leaving the surrounding hypercube.

Definition 2.8 (Rearrangement). Suppose (B_1, C_1) and (B_2, C_2) are packings of the dimension tuples d_1 and d_2 , respectively. The first packing is a *rearrangement* of the second if C_1 coincides with C_2 and if it is possible to continuously move (using only translations and rotations) the general hyperrectangles in B_1 , so that each moved general hyperrectangle coincides with a general hyperrectangle in B_2 without breaking any of the criteria in Definition 2.4 while moving.

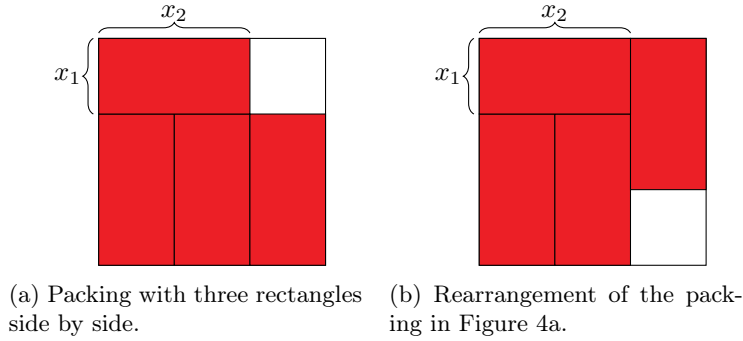


FIGURE 4. Example of equivalent two-dimensional packings of the dimension tuple (x_1, x_2) where $x_2 = 2x_1$.

It is also natural to take reflections and rotations of the packing as a whole into account. We can view such a transformation as a linear map represented by an orthogonal $n \times n$ matrix. There are an infinite number of these, but by Definition 2.4 we only have to consider those mapping the surrounding hypercube to itself. This corresponds to the linear maps sending the standard basis vectors e_1, e_2, \dots, e_n to some permutation of themselves while possibly changing the sign of some of them. Thus, there are $n! \cdot 2^n$ reflections and/or rotations of an n -dimensional hypercube. We will refer to reflections and/or rotations as *symmetries*. With this in mind, we will define what it means for two packings to be equivalent.

Definition 2.9 (Equivalent packings). Suppose there are two packings of the dimension tuples d_1 and d_2 , respectively. These packings are *equivalent* if there exists a rearrangement of the first packing coinciding with a symmetry of the second.

This is an equivalence relation and we will refer to the equivalence classes as unique packings. Notice that it is possible to permute the elements in a dimension tuple without influencing the number of unique packings, since such a permutation corresponds to rotating all packings and thus does not affect the equivalence classes. We can therefore without loss of generality restrict ourselves to increasing dimension tuples. We say that a dimension tuple $d = (x_1, x_2, \dots, x_n)$ is *increasing* if $x_1 \leq x_2 \leq \dots \leq x_n$.

The notion of rearrangements of a packing also gives rise to defining a rigid packing, that is a packing where all the general hyperrectangles are “locked into place” by the surrounding general hyperrectangles or hypercube.

Definition 2.10 (Rigid packing). We say that a packing is *rigid* if all of its rearrangements are identical to itself.

Figure 2 is an example of a rigid packing. Note that the equivalence class of a rigid packing only contains its symmetries, so it contains at most $n! \cdot 2^n$ packings. It might contain fewer if some symmetries are identical to one another.

Next, we will examine packings of dimension tuples satisfying a certain inequality, which forces any packing consisting of hyperrectangles to be rigid and ultimately leads to the definition of a universal packing.

2.2. Universal packings. Before defining a universal packing, it is helpful to get some intuition by looking a bit closer at the two-dimensional packing problem. Consider some increasing dimension tuple (x_1, x_2) . There are ways of organizing the four rectangles, which are only possible when the relative difference between x_1 and x_2 is sufficiently large. Suppose x_2 is twice as large as x_1 , i.e. $x_2 = 2x_1$ or equivalently $x_1 + x_2 = 3x_1$. Then the packing in Figure 4a becomes possible. This packing is only feasible because we can fit three rectangles side by side. It turns out that restricting the search to packings of increasing dimension tuples satisfying the inequality

$$(2) \quad x_1 + x_2 < 3x_1$$

prevents this and restricts the problem in an interesting way. Later, we will argue that any packing of an increasing dimension tuple satisfying (2) consisting of hyperrectangles will be rigid and discuss whether it gives a “recipe” for constructing a packing of *any* dimension tuple as seen in Figure 2. Thus, it is possible to reuse the same “universal packing” even for dimension tuples not satisfying (2). This is the inspiration behind our future definition of a “universal packing”.

Notice that the packing in Figure 4a—which relied on (2) not being satisfied—is not rigid. Intuitively, (2) ensures that the four rectangles resemble a square to such a degree that any packing must necessarily resemble the packing Figure 2b, where four squares are neatly organized in a grid inside the surrounding square. We will return to this concept of a grid shortly. The inequality (2) is inspired by Hoffman who proposed a three-dimensional version of it in [Hof81, p. 215], namely

$$(3) \quad x_1 + x_2 + x_3 < 4x_1.$$

Therefore we will name it after Hoffman when we generalize it to n dimensions.

Definition 2.11 (Hoffman’s inequality). A dimension tuple (x_1, x_2, \dots, x_n) satisfies *Hoffman’s inequality* if it is increasing and

$$x_1 + x_2 + \dots + x_n < (n + 1)x_1.$$

Choosing $x_1 = x_2 = \dots = x_n$ for some positive real number shows that a dimension tuple with this property exists in all dimensions. Notice that (2) and (3) are really Hoffman’s inequality for $n = 2$ and $n = 3$, respectively. Intuitively, Hoffman’s inequality ensures that no packing can contain $n + 1$ or more hyperrectangles side by side. Shortly, we will justify that packings (consisting solely of hyperrectangles) of a dimension tuple satisfying Hoffman’s inequality must be rigid and prove that the hyperrectangles must be arranged in a “grid”.

Hoffman restricts his search for three-dimensional packings in [Hof81, p. 215] to dimension tuples satisfying Hoffman’s inequality for $n = 3$, that is (3). George Miller [Mil06] credits Donald Knuth for searching for three-dimensional packings of the dimension tuple $(3, 4, 5)$ and discovering solutions where one could squeeze an additional 28th 3-dimensional hyperrectangle (brick) into the surrounding cube (which has an extent of $3 + 4 + 5 = 12$). Knuth’s solutions [Knu04] are presented in Figure 5. Notice that in each of them it is possible to remove one of the 28 bricks such that the resulting packing of $(3, 4, 5)$ is not rigid and such that the remaining bricks do not seem to be organized in a “grid”. The attentive reader might have noticed that this particular dimension tuple satisfies $x_1 + x_2 + x_3 = 4x_1$, so it barely breaks with (3), that is Hoffman’s inequality. Thus, Hoffman’s inequality might intuitively be interpreted as a boundary separating us from a whole different class of packings, including those with $n + 1$ or more hyperrectangles side by side. For instance all of the solutions in [Knu04] have four bricks side by side somewhere. As seen in Figure 4a for $n = 2$, this class of packings can not serve

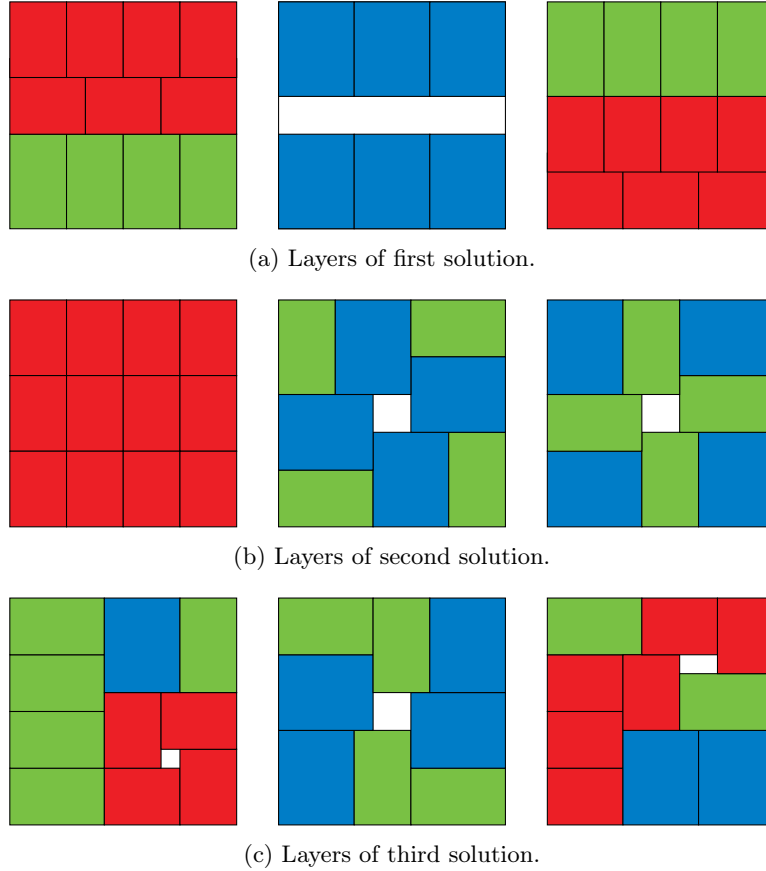


FIGURE 5. The three solutions of the dimension tuple $(3, 4, 5)$ consisting of 28 bricks found by Donald Knuth [Knu04].

as a “recipe” for constructing packings of dimension tuples satisfying Hoffman’s inequality. Therefore we will disregard them in our search for a “universal packing”.

Let us clarify what we mean by a grid. Let $s > 0$ and consider the n -dimensional hypercube $S = (0, s)^n$ which has an extent of s . Let $c_i = is/(n+1)$ for $i = 1, 2, \dots, n$ and observe that $0 < c_i < s$ and also that $c_{i+1} - c_i = s/(n+1)$ for all suitable i . Define $s/(n+1)$ as the *cut distance* and define a *cut* at position k along dimension i to be $C_{i,k} = \{v \in \mathbb{R}^n \mid v_i = c_k\}$ for $i, k = 1, 2, \dots, n$. Intersecting n cuts C_{i,k_i} , one along each dimension $i = 1, 2, \dots, n$, yields a subset of \mathbb{R}^n containing a single point, namely

$$\bigcap_{i=1}^n C_{i,k_i} = \{v \in \mathbb{R}^n \mid v_1 = c_{k_1}, v_2 = c_{k_2}, \dots, v_n = c_{k_n}\} = \{(c_{k_1}, c_{k_2}, \dots, c_{k_n})\},$$

and this point is inside the hypercube S . Any such intersection corresponds to precisely one point in the subset $\{c_1, c_2, \dots, c_n\}^n$ of the hypercube S and we will refer to this subset as the *grid* of the hypercube S and to each of its n^n elements as a *grid point*. Figure 6 illustrates this construction in the three-dimensional case. Using translations, it is always possible to construct the grid of a hypercube. This enables us to formulate and prove the next result. The proof is a formalization and generalization of Hoffman’s proof of the three-dimensional case in [Hof81, p. 218–219] and it is therefore only appropriate to name it after him.

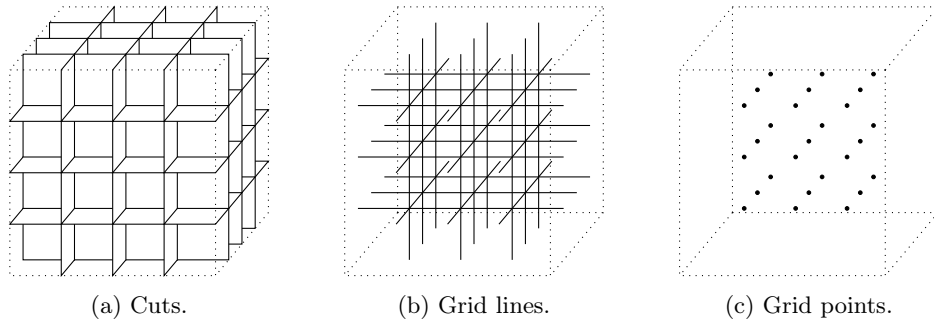


FIGURE 6. Example of cuts and the grid of a cube in three dimensions.

Theorem 2.12 (Hoffman’s grid theorem). Suppose d is a dimension tuple satisfying Hoffman’s inequality and suppose (B, C) is a packing of d such that B contains solely hyperrectangles. Then each hyperrectangle in B contains exactly one grid point of the grid of C .

Proof. Let $d = (x_1, x_2, \dots, x_n)$ and $s = x_1 + x_2 + \dots + x_n$. Assume without loss of generality that the packing has been translated, so that the anchor point of the surrounding hypercube is at the origin of the coordinate system, i.e. $C = (0, s)^n$, whereby we can reuse the grid construction described above.

We first show that every hyperrectangle in B contains a grid point. Take some hyperrectangle R in B and write it as a Cartesian product $R = I_1 \times I_2 \times \dots \times I_n$ where each I_i is a non-empty, open and bounded interval of \mathbb{R} . Next, pick some dimension k in $\{1, 2, \dots, n\}$. We would like to show that there exists a cut along the dimension k which intersects the hyperrectangle R . Let r_k denote the extent of the hyperrectangle R along the k ’th dimension, i.e. the width of the interval I_k . Since (B, C) is a packing of d the hyperrectangle R is a subset of the hypercube C , whereby I_k must be a subset of $(0, s)$. Notice that each c_i from the grid construction is evenly distributed inside $(0, s)$ and cuts it up into $n+1$ intervals of width equal to the cut distance $s/(n+1)$. It then follows from Hoffman’s inequality that $s/(n+1) < x_1 \leq r_k$, so there must exist some j_k in $\{1, 2, \dots, n\}$ such that c_{j_k} is in I_k . Then the cut C_{k, j_k} at position j_k along dimension k must intersect the hyperrectangle R , since all I_i are non-empty. Repeating this procedure for all dimensions k gives n cuts C_{k, j_k} intersecting the hyperrectangle, one along each dimension. Intersecting these cuts results in a set with a single grid point, in this case $(c_{j_1}, c_{j_2}, \dots, c_{j_n})$. This point is inside the hyperrectangle R , since each c_{j_k} is inside I_k . Hence, there is a least one grid point inside each hyperrectangle in B .

Lastly, since (B, C) is a packing of d , then the hyperrectangles in B are pairwise non-overlapping. Then there can not be more than one grid point inside any of the n^n hyperrectangles in B , since there are only n^n grid points in total. ■

Let us introduce a convenient way of working with this grid. Define the n -dimensional grid point coordinates G_n to be the set $\{1, 2, \dots, n\}^n$ and note that for any n -dimensional hypercube, it is possible to identify each of its grid points through the bijective map from G_n to the grid given by $(p_1, p_2, \dots, p_n) \mapsto (c_{p_1}, c_{p_2}, \dots, c_{p_n})$. We will typically refer to a grid point by its coordinates assigned via this bijection.

We say that two grid point coordinates are on the same *grid line* if all but one of their entries match, and we say that this grid line runs along the dimension numbered by the index where the coordinates differ. Note that there are precisely n grid point coordinates on any grid line. For each grid point coordinate p in G_n and each dimension $k = 1, 2, \dots, n$ we define $L(p, k, i)$ as the i ’th grid point coordinate

on the line passing through p along dimension k for $i = 1, 2, \dots, n$, that is

$$L(p, k, i)_j = \begin{cases} i, & \text{if } j = k \\ p_j, & \text{if } j \neq k \end{cases}$$

for $j = 1, 2, \dots, n$ and note that $L(p, k, p_k) = p$ for all $k = 1, 2, \dots, n$.

Suppose $d = (x_1, x_2, \dots, x_n)$ is a dimension tuple satisfying Hoffman's inequality and suppose (B, C) is a packing of d such that B contains solely hyperrectangles. Let p be a grid point coordinate in G_n and for $i = 1, 2, \dots, n$ define $w(p)_i$ as the extent along the i 'th dimension of the hyperrectangle in B containing this particular grid point. This is well-defined by Hoffman's grid theorem 2.12. We measure the *stuffing* on a grid line $L(p, k, i)$ as the sum of the extents along the k 'th dimension of the hyperrectangles associated with the grid points on the grid line, that is

$$\sum_{i=1}^n w(L(p, k, i))_k.$$

A grid line is said to be *completely stuffed* if the stuffing on it equals the extent of the surrounding hypercube. With these notions we are ready to prove the next result, which is a generalization of Hoffman's proof of the three-dimensional case in [Hof81, p. 220] and therefore named after him.

Theorem 2.13 (Hoffman's line theorem). Suppose d is a dimension tuple satisfying Hoffman's inequality and suppose (B, C) is a packing of d consisting solely of hyperrectangles. Then all grid lines are completely stuffed.

Proof. Notice that there are n^n grid lines in total. We wish to measure the stuffing on each of these grid lines. Let S denote the sum of stuffing on all of the n^n grid lines.

By Hoffman's grid theorem 2.12 each of the n^n hyperrectangles contains precisely one grid point. Pick some grid point p , let R_p be the hyperrectangle in B containing p and consider its dimension tuple $r = (r_1, r_2, \dots, r_n)$. Let s denote the sum of d and observe that this dimension tuple must have the same sum as r . The grid point p lies on exactly n grid lines, one along each different dimension. Thus, the hyperrectangle R_p will contribute with r_i worth of stuffing to the grid line through p along dimension i for all $i = 1, 2, \dots, n$. Hence, R_p contributes with a total of s worth of stuffing to S . So does all of the n^n hyperrectangle, whereby $n^n s \leq S$.

Furthermore, none of the n^n grid lines can hold more than s worth of stuffing without breaking some requirement of being a packing as specified in Definition 2.4. Thus, $S \leq n^n s$, so $S = n^n s$, whereby each grid line must have exactly s worth of stuffing. Hence, all grid lines are completely stuffed. ■

This result provides enough information to determine each of the hyperrectangles simply by knowing the dimension tuple of each hyperrectangle, since the only way to fit them inside the surrounding hypercube is by stacking them right next to each other. We can encode all of the information necessary to construct such a packing in a mapping assigning a permutation of $\{1, 2, \dots, n\}$ to each grid point coordinate in G_n specifying the dimension tuple (intuitively "orientation") of the hyperrectangle containing that particular grid point. Let S_n be the set of permutations of $\{1, 2, \dots, n\}$, that is bijections from the set to itself.

Definition 2.14 (Hoffman packing). The map $P: G_n \rightarrow S_n$ is a *Hoffman packing* of a dimension tuple $d = (x_1, x_2, \dots, x_n)$ if the following procedure results in a packing (B, C) of d . Let s denote the sum of d and let the surrounding hypercube $C = (0, s)^n$. For every grid point coordinate p in G_n place a hyperrectangle in B with dimension tuple $w(p) = (x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)})$ where $\sigma = P(p)$ and with anchor point $a(p)$ determined recursively for $i = 1, 2, \dots, n$ by

$$a(p)_i = \begin{cases} 0 & \text{if } p_i = 1 \\ a(L(p, i, p_i - 1))_i + w(L(p, i, p_i - 1))_i & \text{otherwise.} \end{cases}$$

Intuitively, $L(p, i, p_i - 1)$ is the grid point coordinate preceding p on the grid line through p along dimension i . This grid point coordinate exists as long as $1 < p_i \leq n$. The anchor point can also be stated explicitly for $i = 1, 2, \dots, n$ as

$$a(p)_i = \sum_{k=1}^{p_i-1} w(L(p, i, k))_i.$$

We say that P *produces* the packing (B, C) . It is possible to perform the above procedure for any map $P: G_n \rightarrow S_n$ and we will refer to the result as a *pseudo-packing*, since it does not necessarily satisfy the criteria in Definition 2.4 of being a packing of d . For any grid point coordinate p we will refer to the hyperrectangle R_p with anchor-point $a(p)$ and dimension tuple $w(p)$ as the hyperrectangle *associated* with p . Note that if the dimension tuple does not satisfy Hoffman's inequality, then the grid point p need not be inside R_p . We will also refer to the right interval endpoints of R_p as $b(p)$, i.e. $b(p) = a(p) + w(p)$.

Define two Hoffman packings of d as *equivalent* if they produce equivalent packings. We will refer to their equivalence class as a *unique* Hoffman packing. At long last, it is possible to give meaning to a packing which is independent of the choice of dimension tuple.

Definition 2.15 (Universal packing). The map $P: G_n \rightarrow S_n$ is a *universal packing* if it is a Hoffman packing of any dimension tuple satisfying Hoffman's inequality.

This above definition restricts itself to dimension tuples satisfying Hoffman's inequality, but in Section 4.5 we will discuss the possibility of using a universal packing to produce a packing of *any* increasing dimension tuple.

Remark 2.16. When defining a Hoffman packing in Definition 2.14 we have restricted ourselves to packings consisting solely of hyperrectangles (i.e. general hyperrectangles which are in fact hyperrectangles). The primary reason for this restriction is that it turns the packing problem into a combinatorial problem. However, as the focus shifts from Hoffman packings to universal packings we can better justify this choice. A universal packing should in particular produce a packing of a dimension tuple of a hypercube, i.e. a dimension tuple $d = (x_1, x_2, \dots, x_n)$ where $x_1 = x_2 = \dots = x_n$. In this case the AM-GM inequality is in fact an equality, so there can not be any unoccupied "room" in a packing of d . Intuitively, we can convince ourselves that this forces all of the hypercubes to be stacked snugly as Figure 2b illustrates in the two-dimensional case.

We define two universal packings to be *equivalent* if they produce equivalent packings for any dimension tuple satisfying Hoffman's inequality. We will refer to their equivalence class as a *unique* universal packing. Finally, we are able to refine our question of how many different packings there are in a given dimension.

Question 2.17. How many unique universal packings are there in n dimensions?

3. PROPERTIES OF UNIVERSAL PACKINGS

Let us examine the properties of Hoffman packings and universal packings. In [Spi03, p. 1] Spiridonov stipulates that the structure of a three-dimensional packing of a dimension tuple satisfying Hoffman’s inequality must be rigid and notes that “this is probably not necessary; however, it seems tricky to show that no loose packings exist”. We will not attempt to prove this, but we will provide an intuitive argument as to why it is well-founded to claim that a packing of a dimension tuple satisfying Hoffman’s inequality and consisting solely of hyperrectangles is rigid.

Suppose P is a Hoffman packing of a dimension tuple d satisfying Hoffman’s inequality and let (B, C) be the packing of d produced by P . We would like to argue that any rearrangements of this packing is identical to itself. We do so by eliminating the possibility of moving any of the hyperrectangles in B . By Hoffman’s grid theorem 2.12 each of the n^n hyperrectangles contains precisely one grid point. Pick some grid point c with coordinates p and let R_p be the hyperrectangle in B associated with p , i.e. containing c . We begin by ruling out any translations. Consider some dimension k in $\{1, 2, \dots, n\}$ and observe that the hyperrectangles associated with the grid points on the line through p along the k ’th dimension all intersect the line

$$L_k = V_1 \times V_2 \times \dots \times V_n,$$

where $V_k = \mathbb{R}$ and $V_i = \{c_i\}$ for all $i \neq k$. Thus, along all dimensions R_p is wedged between two hyperrectangles (or the surrounding hypercube) due to Hoffman’s line theorem 2.13, see Figure 7b. Intuitively, this is enough to eliminate any continuous translation of the hyperrectangle, since the slightest translation along any dimension will result in the hyperrectangle overlapping with some adjacent hyperrectangle. What about rotations? In Figure 7a it is illustrated how being wedged between two hyperrectangles along all dimensions is not enough to prevent rotations. However, if along all dimensions there is a line running through the hyperrectangle and the adjacent hyperrectangles along that dimension, then the hyperrectangle can not be continuously rotated without overlapping with an adjacent hyperrectangle, see Figure 7b.

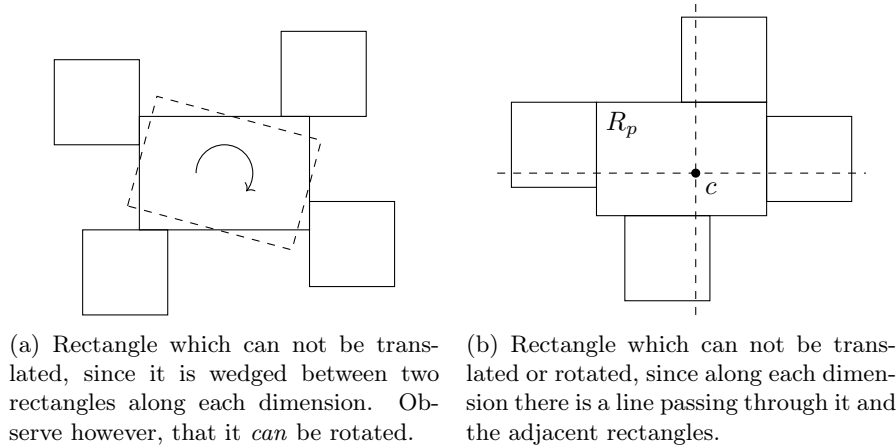


FIGURE 7. Examples illustrating what is necessary to ensure that a hyperrectangle can not be moved.

Lastly, we consider the feasibility of moving multiple hyperrectangles simultaneously. Then in particular we move some specific hyperrectangle R . But moving R forces us to also move some adjacent hyperrectangle to prevent overlap and this

causes a chain reaction resulting in one of the hyperrectangles pricking a hole in the surrounding hypercube C and thus no longer constituting a packing. Hence, without further ado we will state the following proposition.

Proposition 3.1. Suppose d is a dimension tuple satisfying Hoffman's inequality and suppose (B, C) is a packing of d consisting solely of hyperrectangles. Then (B, C) is a rigid packing.

The following result limits the number of universal packings being equivalent.

Lemma 3.2. There exists a strictly increasing sequence of n positive real numbers $x_1 < x_2 < \dots < x_n$ such that (x_1, x_2, \dots, x_n) satisfies Hoffman's inequality and such that the only way to add an arbitrary number of these sequence elements up to $x_1 + x_2 + \dots + x_n$ is to add precisely one of each.

Proof. The case $n = 1$ is trivial, so suppose $n \geq 2$. Define a sequence of n positive real numbers by $x_i = 1 + n^{-(n-i+1)}$ for $i = 1, 2, \dots, n$. Note that this sequence is strictly increasing since

$$x_{i+1} - x_i = n^{-(n-i)} - n^{-(n-i+1)} = n^{-(n-i)}(1 - 1/n) > 0$$

for all suitable i . We now represent each sequence element in base- n . Then the fractional part of x_i consists of $n - i$ zeros followed by a one for all $i = 1, 2, \dots, n$, i.e. $x_n = 1.1_n$, $x_{n-1} = 1.01_n$ and so on. We denote the sum consisting of precisely one of each sequence element by s and we see that

$$s = x_1 + x_2 + \dots + x_n = 10.\overbrace{11\dots 1}_n.$$

Notice that any sum of sequence elements with more than n terms will be strictly greater than s , since the integer part of the sum would be at least 11_n . In particular, the sequence satisfies Hoffman's inequality.

Suppose there exists a sum of sequence elements with n or fewer terms such that they add up to s . Consider the base- n representation of this sum. Precisely one of the terms must be x_1 , since this is the only way to obtain a one at the n 'th position of the fractional part of the sum. Similarly, we can now conclude that precisely one term must be x_2 and using induction the sum must have n terms with precisely one x_i term for each $i = 1, 2, \dots, n$. ■

Lemma 3.3. Suppose d is a dimension tuple with distinct elements. If P_1 and P_2 are Hoffman packings of d producing the same packing, then $P_1 = P_2$.

Proof. Let (B, C) be the packing produced by P_1 and P_2 . Suppose p is a grid point coordinate in G_n . We would like to show that $P_1(p) = P_2(p)$. Consider the dimension tuple r of the hyperrectangle R_p in B associated with p . Since the elements of d are distinct there is only one permutation σ in S_n such that $(d_{\sigma(1)}, d_{\sigma(2)}, \dots, d_{\sigma(n)}) = r$, whereby $P_1(p) = \sigma = P_2(p)$. ■

Proposition 3.4. A unique universal packing contains at most $n! \cdot 2^n$ universal packings.

Proof. Suppose Ω is a unique universal packing. Let d be the dimension tuple consisting of the sequence given by Lemma 3.2. Then all of the Hoffman packings in Ω produce packings of d , which are equivalent. However, none of them are identical due to Lemma 3.3. By Proposition 3.1 these packings are rigid, so there are at most $n! \cdot 2^n$ of them, namely their symmetries. Thus, there are at most $n! \cdot 2^n$ Hoffman packings in Ω . ■

3.1. Verifying a Hoffman packing. When assessing whether a map $P: G_n \rightarrow S_n$ is in fact a universal packing we need to show that it is a Hoffman packing of any dimension tuple d satisfying Hoffman's inequality. Therefore we seek a clever way of ensuring that the pseudo-packing (B, C) produced by P is in fact a packing of d . Hence, we need to make sure that all of the hyperrectangles in B are contained in the surrounding hypercube C and that the hyperrectangles in B are pairwise non-overlapping. Let us begin with a criterion which ensures that the hyperrectangles are contained in the surrounding hypercube.

Criterion 3.5 (Line criterion). Suppose $P: G_n \rightarrow S_n$. Let $d = (x_1, x_2, \dots, x_n)$ be a dimension tuple with distinct elements and consider the pseudo-packing (B, C) of d produced by P . A grid line along some dimension $k \in \{1, 2, \dots, n\}$ is a *unique line* if for all $i = 1, 2, \dots, n$ there is precisely one grid point p on the grid line where the associated hyperrectangle has an extent of x_i along the k 'th dimension, i.e. $P(p)(k) = i$. If all grid lines are unique lines, then P satisfies the *Line criterion*.

Remark 3.6. The Line criterion 3.5 is inspired by Hoffman, who introduced it in the three-dimensional case [Hof81, p. 220]. However, Hoffman's way of justifying it does not scale easily to higher dimensions. However, as we will see below, the notion of a universal packing does in fact allow us to require it in higher dimensions.

It follows from the next result that the Line criterion 3.5 will guarantee the hyperrectangles to be contained in the surrounding hypercube.

Proposition 3.7. Suppose $P: G_n \rightarrow S_n$. Let d be a dimension tuple and let s be its sum. Let (B, C) be the pseudo-packing of d produced by P . If the stuffing on each grid line is no bigger than s , then all of the hyperrectangles in B are contained in the surrounding hypercube C .

Proof. Recall that $C = (0, s)^n$. Suppose p is a grid point coordinate in G_n and let R_p denote the associated hyperrectangle in B . Using the notation from Definition 2.14, we can write R_p as the Cartesian product

$$R_p = \prod_{i=1}^n (a(p)_i, b(p)_i) = (a(p)_1, b(p)_1) \times (a(p)_2, b(p)_2) \times \dots \times (a(p)_n, b(p)_n).$$

Let v be an element in R_p and note that $a(p)_i < v_i < b(p)_i$ for all $i = 1, 2, \dots, n$. In order to show that v lies in C , we need to show that $0 < v_i < s$ for all $i = 1, 2, \dots, n$.

Take some i in $\{1, 2, \dots, n\}$. By examining the definition of the i 'th element of the anchor-point $a(p)$ in Definition 2.14, we observe that $a(p)_i$ is a sum of non-negative real numbers. Hence, $a(p)_i$ is non-negative itself, whereby $0 \leq a(p)_i < v$. Next, consider $b(p)_i = a(p)_i + w(p)_i$, which can be recursively unwrapped into

$$b(p)_i = \sum_{k=1}^{p_i} w(L(p, i, k))_i \leq \sum_{k=1}^n w(L(p, i, k))_i \leq s,$$

since the stuffing on the grid line running through p along dimension i is no bigger than s . Then $v_i < b(p)_i \leq s$, so v is in C . Thus, R_p is contained in C as desired. ■

Next, we show that restricting the search for universal packings to maps $P: G_n \rightarrow S_n$ satisfying the Line criterion 3.5 does not overlook any universal packings.

Proposition 3.8. Suppose P is a universal packing. Then P satisfies the Line criterion 3.5.

Proof. Let $d = (x_1, x_2, \dots, x_n)$ be the dimension tuple consisting of the sequence given by Lemma 3.2 and let s denote its sum. Observe that the elements of d are distinct, since it is strictly increasing. Let (B, C) be the packing of d produced by

P . Take some grid line along some dimension $k \in \{1, 2, \dots, n\}$. We would like to show that this is a unique line. Since d satisfies Hoffman's inequality this grid line is completely stuffed by Hoffman's line theorem 2.13. Recall that by Lemma 3.2 the only way to add the elements of d up to s is to use precisely one of each. There are n grid points on the grid line, so for each $i = 1, 2, \dots, n$ there must be precisely one grid point p_i on the grid line for which the associated hyperrectangle has an extent of x_i along the k 'th dimension, i.e. $P(p_i)(k) = i$. Hence, the grid line is a unique line, whereby P satisfies the Line criterion 3.5. ■

It remains to ensure that the hyperrectangles are pairwise non-overlapping. The naive approach would be to check every pair of hyperrectangles one by one. The next result indicates how such a check could be performed.

Lemma 3.9. Suppose $R_1 = I_1 \times I_2 \times \dots \times I_n$ and $R_2 = J_1 \times J_2 \times \dots \times J_n$ are two n -dimensional hyperrectangles. These two hyperrectangles are overlapping if and only if I_k intersects J_k for all $k = 1, 2, \dots, n$.

Proof. Suppose R_1 and R_2 are overlapping, i.e. they intersect one another. Then there exists a point p in their intersection $R_1 \cap R_2$. Take any k in $\{1, 2, \dots, n\}$. Since p is in R_1 , then p_k is in I_k . Similarly p_k is in J_k . Hence, I_k intersects J_k .

Suppose I_k intersects J_k for all $k = 1, 2, \dots, n$. Then there exists a real number p_k in $I_k \cap J_k$ for all $k = 1, 2, \dots, n$. The point $p = (p_1, p_2, \dots, p_n)$ is in both R_1 and R_2 , so they intersect one another. Hence, R_1 and R_2 are overlapping. ■

Thus, these two hyperrectangles are non-overlapping if and only if there is a k in $\{1, 2, \dots, n\}$ such that I_k and J_k are disjoint. It turns out that it is not necessary to check whether every pair of hyperrectangles are overlapping.

Definition 3.10 (Neighbouring grid points). Two grid points are *neighbours* if their respective coordinates p and q satisfy

$$\max_{i=1}^n |p_i - q_i| = 1.$$

It turns out that it is only necessary to ensure that hyperrectangles associated with neighbouring grid points are non-overlapping.

Criterion 3.11 (No neighbour overlap criterion). Suppose $P: G_n \rightarrow S_n$. Let d be a dimension tuple and let (B, C) be the pseudo-packing of d produced by P . Then P satisfies the *No neighbour overlap criterion* for d if for any pair of neighbouring grid point coordinates p and q in G_n there exists a k in $\{1, 2, \dots, n\}$ with $p_k \neq q_k$ such that

$$\begin{aligned} b(p)_k &\leq a(q)_k && \text{if } p_k < q_k \\ b(q)_k &\leq a(p)_k && \text{if } p_k > q_k, \end{aligned}$$

where $a(p)$ and $a(q)$ are the anchor points, whereas $b(p)$ and $b(q)$ are the right interval endpoints (as defined in Definition 2.14) of the hyperrectangles R_p and R_q associated with p and q , respectively.

The above criterion ensures that the k 'th interval in the Cartesian product of R_p does not intersect the corresponding interval of R_q , whereby the two hyperrectangles are non-overlapping by Lemma 3.9.

Remark 3.12. The No neighbour overlap criterion 3.11 is inspired by Hoffman's observation that a "sharp corner" such as the one in Figure 8a may not occur [Hof81, p. 221]. However, we have found that this concept of a sharp corner extends to higher dimensions, such as the one in Figure 8b. The above criterion encodes the necessary information to detect any type of sharp corner as soon as it emerges.

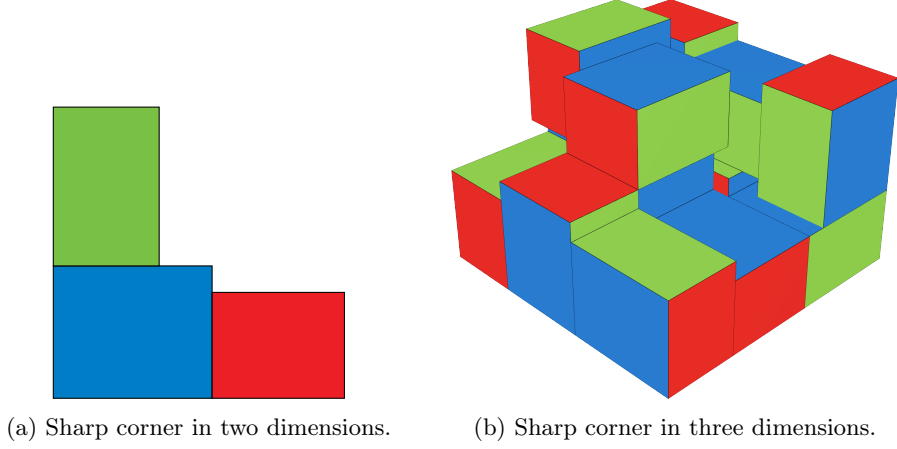


FIGURE 8. Examples of sharp corners. Observe that the three-dimensional sharp corner can not be detected by looking at the squares. Thus, while sharp corners of higher dimensions are theoretically possible, they are not easily visualized.

The next result shows that the No neighbour overlap criterion 3.11 is satisfied by any Hoffman packing of a dimension tuple satisfying Hoffman's inequality.

Proposition 3.13. Suppose $P: G_n \rightarrow S_n$ is a Hoffman packing of a dimension tuple d satisfying Hoffman's inequality. Then P satisfies the No neighbour overlap criterion 3.11 for d .

Proof. Let (B, C) be the packing of d produced by P . Suppose p and q are two neighbouring grid point coordinates in G_n and let

$$R_p = I_1 \times I_2 \times \cdots \times I_n \quad \text{and} \quad R_q = J_1 \times J_2 \times \cdots \times J_n$$

be their associated hyperrectangles in B . Let α and β be the actual grid points corresponding to the coordinates p and q , respectively. By Hoffman's grid theorem 2.12 each grid point lies inside its respective hyperrectangle, i.e. $\alpha \in R_p$ and $\beta \in R_q$. Since (B, C) is a packing R_p and R_q are non-overlapping, so by Lemma 3.9 there exists some k in $\{1, 2, \dots, n\}$ such that I_k does not intersect J_k . We must have that $|p_k - q_k| = 1$, because otherwise $p_k = q_k$, whereby $\alpha_k = \beta_k$ and I_k would intersect J_k . In the notation of the No neighbour overlap criterion 3.11 $I_k = (a(p)_k, b(p)_k)$ and $J_k = (a(q)_k, b(q)_k)$. If $p_k < q_k$, then $\alpha_k < \beta_k$. Then $b(p)_k \leq a(q)_k$ to avoid I_k and J_k intersecting one another. Similarly we conclude for $p_k > q_k$ that $b(q)_k \leq a(p)_k$. Hence, P satisfies the No neighbour overlap criterion 3.11 for d . ■

Next, we show that the No neighbour overlap criterion 3.11 and the Line criterion 3.5 are enough to ensure that a pseudo-packing is in fact a packing. We will need the following result.

Lemma 3.14. Suppose $d = (x_1, x_2, \dots, x_n)$ is a dimension tuple satisfying Hoffman's inequality and suppose A and B are two subsets of $D = \{1, 2, \dots, n\}$. If $|A| < |B|$, then

$$\sum_{i \in A} x_i < \sum_{i \in B} x_i.$$

Proof. Suppose not. Let $C = D \setminus A$. Recall that d is increasing by Hoffman's inequality and note that $|C| = n - |A|$, so $|B| + |C| = |B| + n - |A| \geq n + 1$. Then

$$\begin{aligned} x_1 + x_2 + \cdots + x_n &= \sum_{i \in D} x_i = \sum_{i \in A} x_i + \sum_{i \in C} x_i \\ &\geq \sum_{i \in B} x_i + \sum_{i \in C} x_i \geq (|B| + |C|)x_1 \geq (n + 1)x_1, \end{aligned}$$

which contradicts d satisfying Hoffman's inequality. \blacksquare

Intuitively, the sums consisting of distinct terms from d are ordered by the number of terms in each sum. The situation gets significantly more complicated when two sums consist of the same number of terms, and we will return to this scenario in a moment.

Theorem 3.15. Suppose $P: G_n \rightarrow S_n$ and suppose d is a dimension tuple satisfying Hoffman's inequality. If P satisfies the Line criterion 3.5 and if P satisfies the No neighbour overlap criterion 3.11 for d , then P is a Hoffman packing of d .

Proof. Let (B, C) be the pseudo-packing of d produced by P . We need to show that (B, C) is a packing of d . Let s be the sum of d . Since P satisfies the Line criterion 3.5 the stuffing on each grid line is equal to s , i.e. all grid lines are completely stuffed. By Proposition 3.7, all hyperrectangles in B are contained in the surrounding hypercube C . It only remains to show that the hyperrectangles in B are pairwise non-overlapping.

Let p and q be two grid point coordinates in G_n and let R_p and R_q denote their associated hyperrectangles. Next, consider the value of

$$\delta = \max_{i=1}^n |p_i - q_i|.$$

If $\delta = 0$, then $p = q$ and there is nothing to check. If $\delta = 1$, then p and q are neighbours and it follows by the No neighbour overlap criterion 3.11 that R_p and R_q are non-overlapping. Lastly, suppose $\delta > 1$. Then there exists some k in $\{1, 2, \dots, n\}$ such that $|p_k - q_k| > 1$. Assume without loss of generality that $p_k < q_k$. Consider the k 'th intervals $(a(p)_k, b(p)_k)$ and $(a(q)_k, b(q)_k)$ in the Cartesian products of R_p and R_q , respectively. We would like to show that these do not intersect one another, which can be done by showing that $b(p)_k \leq a(q)_k$. By the construction of (B, C) as described in Definition 2.14 the right endpoint $b(p)_k$ is a sum of p_k terms from d , while the left endpoint $a(q)_k$ is a sum of $q_k - 1$ terms, namely

$$b(p)_k = \sum_{i=1}^{p_k} w(L(p, k, i))_k \quad \text{and} \quad a(q)_k = \sum_{i=1}^{q_k-1} w(L(q, k, i))_k.$$

Since P satisfies the Line criterion 3.5, each term in the sum for $b(p)_k$ can be identified by a distinct index in $\{1, 2, \dots, n\}$. Denote this index set by A and notice that $|A| = p_k$. Similarly we get an index set for B from the sum for $a(q)_k$ and $|B| = q_k - 1$. Note that $|A| = p_k < q_k - 1 = |B|$, so by Lemma 3.14 $b(p)_k < a(q)_k$. Then R_p and R_q are non-overlapping by Lemma 3.9. Thus, every pair of hyperrectangles are non-overlapping, so (B, C) is a packing of d and P is a Hoffman packing of d . \blacksquare

The above proof relies heavily on the dimension tuple d satisfying Hoffman's inequality, where the technicalities have been buried in Lemma 3.14. However, experimental results suggest that we do not need to restrict ourselves to dimension tuples satisfying Hoffman's inequality and we will propose a conjecture in relation to this in Section 4.5.

Remark 3.16. Loosely speaking, Hoffman’s inequality forces sums consisting of distinct terms from the increasing dimension tuple (x_1, x_2, \dots, x_n) to be ordered by the number of terms in each sum (Lemma 3.14). For $n = 3$ this property is actually guarded by the inequality

$$(4) \quad x_3 < x_1 + x_2,$$

which Dean G. Hoffman drew our attention to in recent correspondences [Hof17]. Observe that any dimension tuple (x_1, x_2, x_3) satisfying Hoffman’s inequality will also satisfy (4). However, $d' = (2, 3, 4)$ satisfies (4), but not Hoffman’s inequality. Hence, the conclusion of Lemma 3.14 and Theorem 3.15 should also hold for d' . We suspect the generalization of (4) to higher dimensions to be

$$\sum_{i=\lceil n/2 \rceil + 1}^n x_i < \sum_{i=1}^{\lceil n/2 \rceil} x_i,$$

and we mention this generalized inequality for future studies, because we suspect it to be a more precise requirement for a packing produced by a Hoffman packing to be rigid, i.e. without any “loose bricks”.

3.2. Promoting a Hoffman packing to a universal packing. By Theorem 3.15 we can guarantee that a map $P: G_n \rightarrow S_n$ is a universal packing by making sure that it satisfies the Line criterion 3.5 and that it satisfies the No neighbour overlap criterion 3.11 for any choice of dimension tuple satisfying Hoffman’s inequality. This is a lot of dimension tuples to examine and gives rise to the following definition.

Definition 3.17 (Representative dimension tuple set). Suppose T is a set of increasing dimension tuples. Then T is a *representative dimension tuple set* (RDTS) if it has the following property: Suppose $P: G_n \rightarrow S_n$ satisfies the Line criterion 3.5 and satisfies the No neighbour overlap criterion 3.11 for all dimension tuples in T , then P satisfies the No neighbour overlap criterion 3.11 for *any* increasing dimension tuple.

We can restrict ourselves to maps satisfying the Line criterion 3.5, since it is a requirement for being a universal packing by Proposition 3.8. Observe that the set of all increasing dimension tuples constitute a RDTS. However, it turns out that far fewer dimension tuples are required to constitute a RDTS. This greatly simplifies the practical search for universal packings, since it enables us to search for universal packings using just a few specific dimension tuples.

Suppose $P: G_n \rightarrow S_n$ is a map satisfying the Line criterion 3.5. If P satisfies the No neighbour overlap criterion 3.11 for all dimension tuples in a RDTS, then it will in fact be a universal packing by Theorem 3.15. When constructing a RDTS T the strategy will be to include just enough dimension tuples to force P to satisfy the No neighbour overlap criterion 3.11 for any choice of increasing dimension tuple as long as P satisfies this criterion for the dimension tuples in T .

Suppose $d = (x_1, x_2, \dots, x_n)$ is an increasing dimension tuple and let us examine the No neighbour overlap criterion 3.11 in more depth. For each pair of neighboring grid point coordinates p and q in G_n the No neighbour overlap criterion 3.11 considers a number of inequalities, namely one for each entry where p and q differ. We will refer to such a composition of inequalities as an *overlap comparison* and to each of the inequalities as an *overlap inequality*. Consider an overlap inequality from the overlap comparisons of p and q . Notice that it has the same number of terms t from d on each side of the inequality sign and that $0 < t < n$. We will refer to the positive integer t as the *arity* of the overlap inequality. Consider the terms on one side. Since P satisfies the Line criterion 3.5, then each term x_i on this

side is identified by a unique i in $\{1, 2, \dots, n\}$. Hence, we can identify an overlap inequality in the following way.

Definition 3.18 (Overlap inequality). Suppose A and B are two non-empty proper subsets of $\{1, 2, \dots, n\}$ with the same cardinality $|A| = |B| = t$. Then (A, B) is an n -dimensional *overlap inequality* with *arity* t .

We say that the dimension tuple d *satisfies* the overlap inequality if

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i.$$

Be aware that for the sake of readability, we might sometimes write an overlap inequality as above, even if each x_i have not been defined.

Some overlap inequalities are satisfied for any choice of increasing dimension tuple. Suppose (A, B) is an n -dimensional overlap inequality with arity t . Let $(\alpha_i)_{i=1}^t$ and $(\beta_i)_{i=1}^t$ be the strictly increasing sequences of element in A and B , respectively. If $\alpha_i \leq \beta_i$ for all $i = 1, 2, \dots, t$, then the overlap inequality is satisfied by any increasing dimension tuple. We will refer to such an overlap inequality as *trivial*. It is convenient to introduce the following equivalence relation between n -dimensional increasing dimension tuples. Suppose $d_1 = (x_1, x_2, \dots, x_n)$ and $d_2 = (y_1, y_2, \dots, y_n)$ are two increasing dimension tuples. We define these dimension tuples to be equivalent if they satisfy the same overlap inequalities, that is for any non-empty proper subsets A and B of $\{1, 2, \dots, n\}$ with $|A| = |B|$, then

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i \Leftrightarrow \sum_{i \in A} y_i \leq \sum_{i \in B} y_i.$$

We denote the equivalence class of d_1 by $[d_1]$. Then we can establish a well-defined partial ordering on the equivalence classes, where $[d_1] \leq [d_2]$ if any overlap inequality satisfied by d_1 is also satisfied by d_2 , that is

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i \Rightarrow \sum_{i \in A} y_i \leq \sum_{i \in B} y_i.$$

Notice that the equivalence class $[(x_1, x_2, \dots, x_n)]$ where $x_1 = x_2 = \dots = x_n$ containing the dimension tuple of any hypercube is the greatest element with respect to this ordering, since it satisfies all overlap inequalities. The next result shows how this partial ordering enables us to propagate information about the No neighbour overlap criterion 3.11.

Proposition 3.19. Suppose d_1 and d_2 are two increasing dimension tuples with $[d_1] \leq [d_2]$. If $P: G_n \rightarrow S_n$ satisfies the Line criterion 3.5 and satisfies the No neighbour overlap criterion 3.11 for d_1 , then it does so for d_2 as well.

Proof. Since P satisfies the Line criterion 3.5 all overlap comparisons consist of inequalities which can be identified as in Definition 3.18. Since $[d_1] \leq [d_2]$ the overlap inequalities satisfied by d_1 are satisfied by d_2 as well. Then P satisfies the No neighbour overlap criterion 3.11 for d_2 . ■

The construction of these equivalence classes enables us to show the following result.

Proposition 3.20. The number of equivalence classes of increasing dimension tuples is finite.

Proof. Let us begin by establishing an upper bound on the number of different overlap inequalities. First, we count the number of overlap inequalities with a certain arity t and note that $0 < t < n$. The number of subsets of

$\{1, 2, \dots, n\}$ with t elements is $\binom{n}{t}$, so the number of overlap inequalities with arity t is $\binom{n}{t}^2$. Hence the total number of overlap inequalities is

$$k = \sum_{i=1}^{n-1} \binom{n}{i}^2.$$

When determining which equivalence class an increasing dimension tuple belongs to it depends on which overlap inequalities are satisfied and which are not. Hence, the number of equivalence classes is no greater than 2^k . ■

This information implies that there exists a finite RDTS in any dimension n , since we can construct such a set by picking a representative from each of the finitely many equivalence classes. We can actually show that any equivalence class has a representative satisfying Hoffman's inequality. Having a RDTS consisting of dimension tuples all satisfying Hoffman's inequality simplifies the search for a universal packing due to Proposition 3.13.

Proposition 3.21. Suppose d is an increasing dimension tuple. Then there exists a dimension tuple d' satisfying Hoffman's inequality and such that $[d] = [d']$.

Proof. Let $d = (x_1, x_2, \dots, x_n)$ and let $c = \sum_{i=1}^n (x_i - x_1)$ and note that $c \geq 0$. Let $d' = (y_1, y_2, \dots, y_n)$ where $y_i = x_i + c$ for all $i = 1, 2, \dots, n$. Note that for any overlap inequality (A, B)

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i \Leftrightarrow \sum_{i \in A} (x_i + c) \leq \sum_{i \in B} (x_i + c) \Leftrightarrow \sum_{i \in A} y_i \leq \sum_{i \in B} y_i,$$

since $|A| = |B|$, whereby $[d] = [d']$. Next, we see that

$$\sum_{i=1}^n (y_i - y_1) = \sum_{i=1}^n (x_i - x_1) = c < x_1 + c = y_1,$$

and it follows that d' satisfies Hoffman's inequality as desired. ■

In addition, we can do better in order to reduce the number of dimension tuples necessary to constitute a RDTS.

Proposition 3.22. Suppose T is a set of increasing dimension tuples such that T contains a representative from each minimal equivalence class. Then T is a RDTS.

Proof. Suppose $P: G_n \rightarrow S_n$ satisfies the Line criterion 3.5 and satisfies the No neighbour overlap criterion 3.11 for all dimension tuples in T . Suppose d is an increasing dimension tuple. By the construction of T there must be some dimension tuple d' in T such that $[d'] \leq [d]$. Since P satisfies the No neighbour overlap criterion 3.11 for d' , then by Proposition 3.19 it does so for d as well. Hence, T is a RDTS. ■

Thus, if we can show that some equivalence class is in fact the least element, then the set consisting of just one of the dimension tuples from this equivalence class will constitute a RDTS. Next, we show a result which simplifies determining a small RDTS, and give an example of using it in the two-dimensional case.

Proposition 3.23. Suppose d is an increasing dimension tuple with distinct elements. Then any overlap inequality with arity 1 satisfied by d is trivial.

Proof. Let $d = (x_1, x_2, \dots, x_n)$ and observe that it must be strictly increasing. Suppose (A, B) is an overlap inequality with arity 1, i.e. A and B are non-empty proper subsets of $\{1, 2, \dots, n\}$ with $|A| = |B| = 1$. Then we can write $A = \{i\}$ and $B = \{j\}$ for some i and j in $\{1, 2, \dots, n\}$. If $i \leq j$, then the overlap inequality is trivial. Suppose not, i.e. $i > j$. Then $x_i \leq x_j$, which contradicts d being strictly increasing. ■

Example 3.24 (RDTS for $n = 2$). In this case all overlap inequalities have arity 1. For an increasing dimension tuple (x_1, x_2) there are four overlap inequalities,

$$x_1 \leq x_1, \quad x_1 \leq x_2, \quad x_2 \leq x_1 \quad \text{and} \quad x_2 \leq x_2.$$

Three of these are trivial, while $x_2 \leq x_1$ —strictly speaking identified by the tuple $(\{2\}, \{1\})$ —is the only non-trivial overlap inequality. Observe that $d = (2, 3)$ satisfies Hoffman’s inequality and contains distinct elements. Then d satisfies only trivial overlap inequalities by Proposition 3.23. Hence $[d]$ is the least element, so by Proposition 3.22

$$T = \{(2, 3)\}$$

is a RDTS for $n = 2$.

Before considering the three-dimensional case, we will show that an overlap inequality with sufficiently high arity is actually an overlap inequality with a lower arity in disguise. Intuitively, it has so many terms on each side of the inequality sign that some identical terms must occur on both sides. Subtracting these terms from both sides yields an overlap inequality with lower arity, which is satisfied precisely when the original overlap inequality is satisfied.

Lemma 3.25. Suppose (A, B) is a non-trivial n -dimensional overlap inequality with arity s . Then there exists an overlap inequality (A', B') with arity $s - |A \cap B|$ which is satisfied precisely when (A, B) is satisfied.

Proof. Note that $A \neq B$, since the overlap inequality is non-trivial. Let $C = A \cap B$, $A' = A \setminus C$ and $B' = B \setminus C$. Then

$$\sum_{i \in A} x_i \leq \sum_{i \in B} x_i \Leftrightarrow \sum_{i \in A'} x_i + \sum_{i \in C} x_i \leq \sum_{i \in B'} x_i + \sum_{i \in C} x_i \Leftrightarrow \sum_{i \in A'} x_i \leq \sum_{i \in B'} x_i$$

for any increasing dimension tuple $d = (x_1, x_2, \dots, x_n)$. Finally, note that (A', B') has arity $t = s - |A \cap B|$ and that $0 < t$, since $|A \cap B| < s$. ■

Proposition 3.26. Suppose (A, B) is an n -dimensional overlap inequality. Then there exists an overlap inequality (A', B') with arity t such that $0 < t \leq n/2$ and which is satisfied precisely when (A, B) is satisfied.

Proof. If $n = 1$ there are no overlap inequalities, so suppose $n \geq 2$. Let s be the arity of (A, B) , that is $s = |A| = |B|$. If $s \leq n/2$ we are done by choosing (A', B') to be (A, B) . Suppose $s > n/2$. If $A = B$, then the overlap inequality is trivial and we can simply choose $(A', B') = (\{1\}, \{1\})$. Otherwise by Lemma 3.25 there exists an overlap inequality (A', B') with arity $t = s - |A \cap B|$ which is satisfied precisely when (A, B) is satisfied. We claim that $t \leq n/2$. Otherwise $n/2 < t$ and

$$n < 2t = 2s - 2|A \cap B| = |A| + |B| - 2|A \cap B| = |A \cup B| - |A \cap B| \leq |A \cup B|,$$

which is impossible as A and B are subsets of $\{1, 2, \dots, n\}$ with only n elements. ■

Example 3.27 (RDTS for $n = 3$). In this case all overlap inequalities have arity 1 or 2. Observe that $d = (4, 5, 6)$ satisfies Hoffman’s inequality and contains distinct elements. Of the overlap inequalities with arity 1, then d satisfies only the trivial ones by Proposition 3.23. We will show that the same is the case for overlap inequalities with arity 2. Suppose (A, B) is an overlap inequality with arity 2 which is satisfied by d . Then by Proposition 3.26 there exists an overlap inequality (A', B') with arity 1 which is also satisfied by d . But then (A, B) is trivial itself. Hence d satisfies only trivial overlap inequalities, so $[d]$ is the least element and by Proposition 3.22

$$T = \{(4, 5, 6)\}$$

is a RDTS for $n = 3$. Hence, we can use this specific dimension tuple when searching for three-dimensional universal packings.

4. EXPERIMENTS IN THE THREE-DIMENSIONAL CASE

Let us attempt to find all of the three-dimensional universal packings using a computer program. We seek an algorithm which outputs the maps from G_3 to S_3 which are universal packings. We will refer to such a map as a solution. However, in order to count the number of unique universal packings we will not count a solution if we have already counted an equivalent solution. In the three-dimensional case we will refer to the hyperrectangles as bricks. We will approach this task using two different approaches, namely dynamic programming/memoization and backtracking.

4.1. Dynamic programming. The idea behind dynamic programming [CLRS09, p. 359] is to solve a problem by combining the solutions of a number of simpler subproblems. We solve each of these subproblems once and store their solutions. Whenever one of these subproblems occurs again, then instead of recomputing its solution, we will simply look it up. Thus, dynamic programming uses additional memory to save computation time. The idea of storing solutions to the subproblems instead of recomputing them is called “memoization” [CLRS09, p. 365].

In order to apply this method to our problem, we need to specify what our subproblems are. This leads to the definition of a subgrid.

Definition 4.1 (Subgrid). Suppose U is a subset of $G_n = \{1, 2, \dots, n\}^n$ obtained by fixing k entries of the elements in G_n and let $m = n - k$. We say that U is an m -dimensional *subgrid* of G_n .

Observe that a one-dimensional subgrid corresponds to a grid line. Suppose U is an m -dimensional subgrid of G_n and suppose $P: U \rightarrow S_n$. Given some n -dimensional dimension tuple, we can—without going into details—extend the procedure from the definition of a Hoffman packing to produce what we will refer to as a *subpacking* consisting of m -dimensional hyperrectangles. It is also straight forward to extend the notion of equivalent subpackings. We will refer to a subpacking as a line, square and cube if $m = 1, 2$ and 3 , respectively.

The first idea that springs to mind for applying dynamic programming to this problem is to first figure out every possible line and store these. Then use these lines to construct every possible square and store them. Finally, we use these squares to construct all possible cubes, that is all solutions to our problem.

Observation 4.2. Using a computer program we have produced all lines, squares and cubes of the dimension tuple $(4, 5, 6)$. The results are presented in Table 1. There is one subtlety in relation to combining solutions of the subproblems. When we have constructed a line there are two options left for the width of each line segment when combining lines to construct a square. We will refer to this as *specialization*. There are three line segments on a line, and hence there are really $6 \cdot 2^3 = 48$ specialized lines. Given a square there is only one option left for which height to assign each rectangle. Thus, in the three-dimensional case we can think of a square as already specialized.

TABLE 1. Number of lines, squares (satisfying the Subgrid criterion 5.10) and cubes of the dimension tuple $(4, 5, 6)$. Numbers in parentheses state the counts ignoring symmetries.

Lines	Squares	Cubes
6 (3)	624 (78)	1 008 (21)

4.2. Backtracking. Backtracking is a method for solving a problem by incrementally constructing a candidate solution, but “backtracking” when we can predict that this candidate can not possibly be completed to a solution. The usefulness of this technique relies on being able to predict failure early on and that such a test is relatively quick to perform. This enables us to efficiently traverse the search tree and prune the branches along the way.

The immediate way to apply backtracking to this problem is by placing one brick at a time and every time testing whether it might still be possible to complete the partial candidate to a solution. If there is still hope, we carry on and place another brick. If not, we remove the latest brick and register this rejection. If all orientations of a brick have been rejected, we backtrack, that is we remove the second latest brick and register this rejection. If all $3^3 = 27$ bricks have been successfully placed we save the solution. In order to continue searching for more solutions we can remove the last brick and artificially register this as a rejection.

In practise we represent a brick by its anchor point and dimension tuple. Placing a brick corresponds to inserting this information into a three-dimensional array structure where each array index represents a grid point coordinate. In order to determine and assign an anchor point to the brick, we need the bricks “before” it to be assigned beforehand. We can ensure this by considering the lexicographical ordering of the grid point coordinates and placing the bricks according to this ordering. During the course of this project, it has been possible to gradually improve

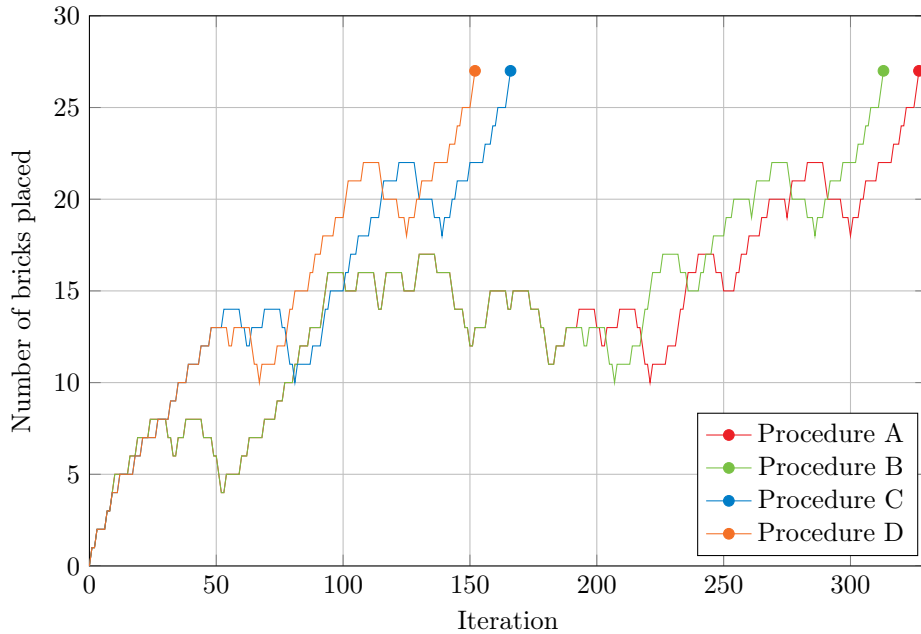


FIGURE 9. Plot of the number of bricks placed after a certain number of iterations in the three-dimensional case depending on the different test procedures. Here, we terminate as soon as a solution is found, i.e. 27 bricks have been successfully placed.

the predictive power of the procedure which tests whether a partial candidate can not possibly be completed into a solution. The four procedures tried out are described in Table 2. Procedure D is a notable improvement over the naive procedure A requiring 53% fewer iterations to find a single solution and 42% fewer iterations to traverse the entire search tree and find all solutions. The details can be found

TABLE 2. Description of the four procedures used to predict whether a partial candidate can not possibly be completed.

Procedure	Description
A	Checks whether the Line criterion 3.5 is violated or whether two neighbouring bricks are overlapping.
B	Checks all from procedure A and also whether there are sharp corners like in Figure 8.
C	Checks all from procedure A and also whether the Subgrid criterion 5.10 is violated.
D	Checks all of the above.

in Table 3. This improvement and the significance of the different criteria is illustrated in Figure 9. While the running times of the procedures are different, we choose to compare number of iterations, since this gives a better indication of the branch pruning capabilities.

TABLE 3. Number of iterations needed to find a single solution and all solutions, respectively, in the three-dimensional case depending on the different test procedures.

Procedure	A	B	C	D
One solution	327	313	166	152
All solutions	646 009	500 185	471 569	371 560

4.3. Results and analysis of solutions. By Example 3.27 the set $\{(4, 5, 6)\}$ is a RDTS for $n = 3$, so it follows by Theorem 3.15 that any unique Hoffman packing of $(4, 5, 6)$ satisfying the Line criterion 3.5 is in fact a unique universal packing.

Observation 4.3. Using the dynamic programming and backtracking approaches both yielded the same number of unique Hoffman packings of the dimension tuple $(4, 5, 6)$, namely 21. These are presented in Appendix A. Thus, the answer to Question 2.17 is 21 for $n = 3$. In fact, this enables us to give the first proof that 3 is a good dimension in Corollary 4.7.

We can define the *distance* between two universal packings P_1 and P_2 as the smallest number of function values of P_1 which we need to alter (intuitively, the smallest number of hyperrectangles we need to modify) such that the resulting map is equivalent to P_2 for any dimension tuple satisfying Hoffman’s inequality. This defines a metric l on the unique universal packings.

Observation 4.4. We have computed the distances between representatives from each of the 21 unique universal packings in the three-dimensional case. It turns out that the distance between any two non-equivalent universal packings P_1 and P_2 satisfies $6 \leq l(P_1, P_2) \leq 17$. In fact for any universal packing P , there exists a non-equivalent universal packing P' such that $l(P, P') = 6$. Intuitively, we can always remove up to 5 arbitrary bricks from a universal packing and still have only one way to complete this partial solution.

The importance of the above observation is unclear. However, we mention it as we speculate that a future and more thorough examination might reveal more knowledge of the problem. It might also be rewarding to compare it to the analysis of the 21 unique universal packings in [BCG04, p. 913–915].

Suppose σ is a permutation in S_n . We say that a universal packing P is *stable* under the permutation σ if $P': G_n \rightarrow S_n$ defined by

$$P'(p) = P(p) \circ \sigma$$

for all p in G_n is a universal packing. Note that we can determine whether a unique universal packing exhibits such a property by examining a representative.

Observation 4.5. We have examined whether the 21 unique universal packings in the three-dimensional case are stable under any permutation from S_3 . Trivially, all 21 universal packings are stable under the identity permutation. In addition, 17 (numbered 1–17 in Appendix A) of the unique universal packings are stable under the reversing permutation, i.e.

$$\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}.$$

Note that performing this operation twice, will undo it, since σ^2 is the identity. This operation associates 16 of the unique universal packings in pairs, while one of the unique universal packings exhibits the remarkable property that performing this operation on a representative yields a universal packing, which is equivalent to the representative. This particular packing can be seen in Figure 3. These results are consistent with [Spi03, p. 5] and [BCG04, p. 914] where the phenomenon is described as “duality” of packings. Interestingly, by looking at the packings in Appendix A we have also noticed that all of these packings contain the square seen in Figure 3a and we have checked that the remaining packings 18–21 do not. Also, we found that no universal packing is stable under any other permutation from S_3 .

The above observation shows that “duality” is not a necessary condition for being a universal packing, but we speculate that it might be possible to exploit this phenomenon to improve the branch pruning when searching for a solution in higher dimensions. Requiring this property and using procedure D, we found the 17 unique universal packings in 349 027 iterations, which is a reduction by 6%. However, it did not reduce the number of iterations needed to find the first solution.

4.4. Comparison of the two approaches. Both implementations take a split second to complete their search. Let us consider the advantages of the backtracking approach as opposed to dynamic programming. First of all, backtracking is well suited for showing the existence of a solution, since it performs only the work needed to obtain it. This is certainly not the case for the gradual solving in the bottom-up dynamic programming approach, where all lines are produced, then all squares and so on. Also, the backtracking approach has a tiny memory footprint, since it does not store solutions to subproblems. This will also let the implementation take better advantage of the caching capabilities in modern computer processors.

Secondly, with backtracking it is possible to discard symmetries, i.e. different equivalent packings, early on. If n is odd then any universal packing will have a hyperrectangle in its “middle” and we can discard all rotations by fixing this hyperrectangle. If n is even then we can still somewhat reduce the number of symmetries by fixing a group of hyperrectangles around the “middle”, but—as we will see in the four-dimensional case—we need to be careful to get this right. It is not clear how one should go about doing something similar using dynamic programming, since we do not know which symmetries of a subproblem might be needed to solve the more complex problem beforehand. Hence, we can not discard these until the end.

However, the backtracking approach has one major drawback, namely that it is very difficult to say something meaningful about its progress. Using dynamic programming it is possible to provide a somewhat reasonable estimate on the amount of work left as we will provide in the four-dimensional case later.

Another advantage of the dynamic programming approach is that it is well suited for parallelization, since subproblems of the same size can be solved independently. For instance multiple threads can work together to construct all squares. However, as we will see shortly in the four-dimensional case, attempting to reduce the number of symmetries naturally divides the problems into independent problems, which the backtracking approach might be best suited for.

4.5. Producing a packing of any dimension tuple using universal packing.

The motive behind introducing the notion of a universal packing was twofold. First, we wanted the number of packings to be independent of the choice of dimension tuple; and secondly, because for $n \leq 3$ such a packing actually provides a “recipe” for constructing a packing of *any* increasing dimension tuple, thereby proving these dimensions to be good. We put forward the following conjecture.

Conjecture 4.6. Suppose $P: G_n \rightarrow S_n$ is a universal packing. Then P produces a packing of any increasing dimension tuple. In particular n is a good dimension.

The truth of this conjecture is not too important for $n = 4$, since we know 4 to be a good dimension by Theorem 2.7. However, it would enable us to relax the definition of a universal packing to better suit its name and also further justify our formulation of Question 2.17. But most importantly, for $n = 5$ and larger primes, this would help pave the way of showing such a dimension to be good. First, one would determine a RDTS of dimension tuples satisfying Hoffman’s inequality, then find a Hoffman packing of all these (and satisfying the Line criterion 3.5). Finally, one would promote it to be a universal packing by Theorem 3.15 and the dimension of interest would be good by the conjecture above.

Proving the above conjecture loosely speaking comes down to showing that any pair of hyperrectangles in a pseudo-packing are non-overlapping under the assumption of the Line criterion 3.5 and that any pair of hyperrectangles associated with neighbouring grid points are non-overlapping. Intuitively, this seems obvious. However, when producing a pseudo-packing of a dimension tuple not satisfying Hoffman’s inequality, then the hyperrectangle associated with a grid point does not necessarily contain this grid point. Hence, the intuition of “neighbouring hyperrectangles” might be deceiving. Apart from this intuition, the basis for proposing this conjecture is that we prove it for $n \leq 3$ below and the failed attempt to disprove it in the four-dimensional case which is documented in Observation 5.4.

The following proof works for $n \leq 3$, but it does not seem very straightforward to generalize it to higher dimensions, since even the case $n = 3$ is quite tedious.

Proof of Conjecture 4.6 for $n \leq 3$. The case $n = 1$ is trivial. If $n = 2$ the only universal packing (up to equivalence) is the one illustrated in Figure 2, which clearly produces a packing of any increasing dimension tuple. Suppose $n = 3$ and suppose $d = (x_1, x_2, x_3)$ is an increasing dimension tuple and let (B, C) be the pseudo-packing of d produced by P . Note that P satisfies the Line criterion 3.5 by Proposition 3.8. Note that by Proposition 3.21 there exists a RDTS for consisting of dimension tuples satisfying Hoffman’s inequality (for instance the one presented in Example 3.27). Hence, P satisfies the No neighbour overlap criterion 3.11 for each of these dimension tuples by Proposition 3.13. Since they constitute a RDTS, then P must also satisfy this criterion for d .

Notice that the Line criterion 3.5 guarantees that the hyperrectangles in B will be contained in the surrounding hypercube C by Proposition 3.7. Next, we show

that the hyperrectangles are pairwise non-overlapping. Take two distinct grid point coordinates p and q in G_3 and let R_p and R_q denote their associated bricks in B . Let

$$\delta = \max_{i=1}^n |p_i - q_i|,$$

and note that δ is either 1 or 2. If $\delta = 1$, then p and q are neighbours and we are done due to the No neighbour overlap criterion 3.11 being satisfied for d . Suppose $\delta = 2$. Now, we assume that $p_i < q_i$ for all suitable i and consider only the following case of $\Delta = q - p$, namely

$$(2, 0, 0), \quad (2, 1, 0), \quad (2, 1, 1), \quad (2, 2, 0), \quad (2, 2, 1) \quad \text{and} \quad (2, 2, 2).$$

Observe that we can do so without loss of generality, since we can always reflect and/or rotate the packing such that we end up in one of the above cases.

In the following we use the same notation as in the construction of a Hoffman packing, i.e. for a grid point g in G_3 , then $a(g)$ and $b(g)$ are the left and right interval endpoints of the associated hyperrectangle R_g and $w(g) = b(g) - a(g)$ is the dimension tuple of the hyperrectangle R_g .

First, observe that for R_p and R_q to be overlapping along a dimension $k \in \{1, 2, 3\}$ where $|p_k - q_k| = 2$, then $w(p)_k = x_3 = w(q)_k$. Otherwise if for instance $w(q)_k$ is either x_1 or x_2 then

$$b(p)_k = w(p)_k \leq x_3 \leq x_1 + x_2 + x_3 - w(q)_k = a(q)_k.$$

Hence, if we are in one of the three last case there can not be any overlap, since p and q differ by 2 along two dimensions. Also, the first case $\Delta = (2, 0, 0)$ is straightforward, since $b(p)_1 \leq a(q)_1$ by construction of the pseudo-packing.

Next consider $\Delta = (2, 1, 0)$ and suppose for contradiction that R_p and R_q are overlapping. Then in particular

$$a(q)_1 < b(p)_1 \quad \text{and} \quad a(q)_2 < b(p)_2.$$

It is helpful to glance at Figure 10 for this next bit. Consider the grid point coordinate $r = p + (1, 1, 0) = q + (-1, 0, 0)$. By the No neighbour overlap criterion 3.11

$$b(p)_1 \leq a(r)_1 \quad \text{or} \quad b(p)_2 \leq a(r)_2,$$

but $b(p)_1 \leq a(r)_1$ can not be the case, since $a(r)_1 \leq b(r)_1 = a(q)_1 < b(p)_1$. Then $b(p)_2 \leq a(r)_2$ must be the case. Next, consider the grid point coordinate $s = p + (1, 0, 0) = q + (-1, -1, 0)$ and note that $b(s)_2 = a(r)_2$. Now, notice that

$$a(q)_1 < b(p)_1 = a(s)_1 \leq b(s)_1 \quad \text{and} \quad a(q)_2 < b(p)_2 \leq a(r)_2 = b(s)_2,$$

which contradicts P satisfying the No neighbour overlap criterion for s and q . Finally, consider the case $\Delta = (2, 1, 1)$ and suppose for contradiction that R_p and R_q are overlapping. Note that R_p and R_q must be positioned in one of the two ways illustrated in Figure 11. First, consider the case illustrated in Figure 11a where R_p is in a corner and R_q is in the middle of the opposite side. Note that $w(p)_1 = w(q)_1 = x_3$ for these bricks to overlap along the x -axis. Then for the bricks to overlap along the y -axis $w(p)_2 = x_2$ and $a(q)_2 = x_1$. But then $w(p)_3 = x_1$ and R_p can not be overlapping with R_q , since there is a brick below R_q with a height of at least x_1 .

Next, consider the case illustrated in Figure 11b where R_p and R_q are each positioned in the middle of an edge. Again, along the x -axis $w(p)_1 = w(q)_1 = x_3$ for these bricks to overlap. Then for the bricks to overlap along the y -axis $w(q)_2 = x_2$ and the brick in front of R_p must have an extent of x_1 along the y -axis. But then $w(p)_2 = x_2$, so $w(p)_3 = x_1$ and again R_p can not be overlapping with R_q , since there is a brick below R_q with a height of at least x_1 . Hence, the bricks are pairwise non-overlapping, whereby (B, C) is a packing of d as desired. ■

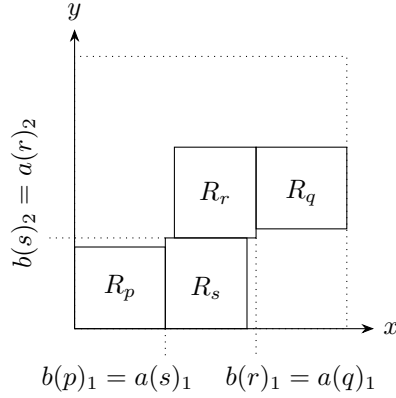
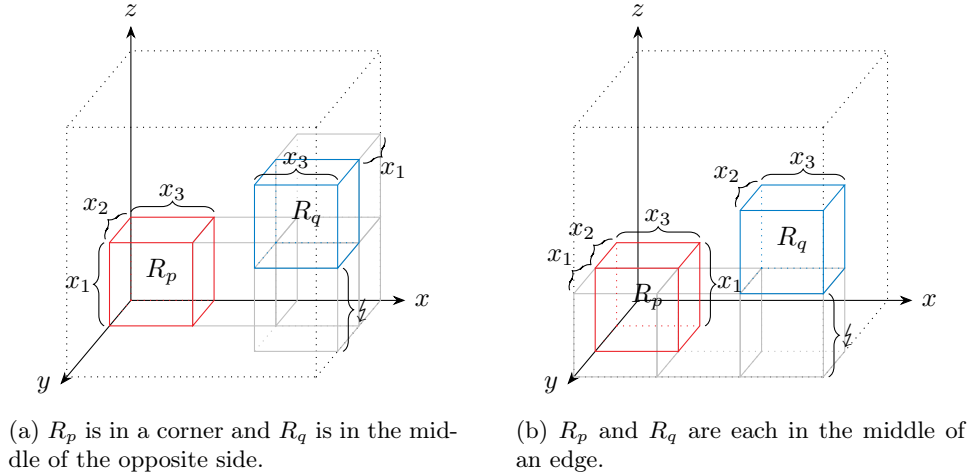


FIGURE 10. Abstract relations between the interval endpoints of the bricks in the case $\Delta = (2, 1, 0)$. This figure is highly distorted, since R_p and R_q are assumed to be overlapping in the proof.



(a) R_p is in a corner and R_q is in the middle of the opposite side.

(b) R_p and R_q are each in the middle of an edge.

FIGURE 11. The two possible constellations of the bricks R_p and R_q in the case $\Delta = (2, 1, 1)$. The figures are highly distorted, since R_p and R_q are assumed to be overlapping in the proof.

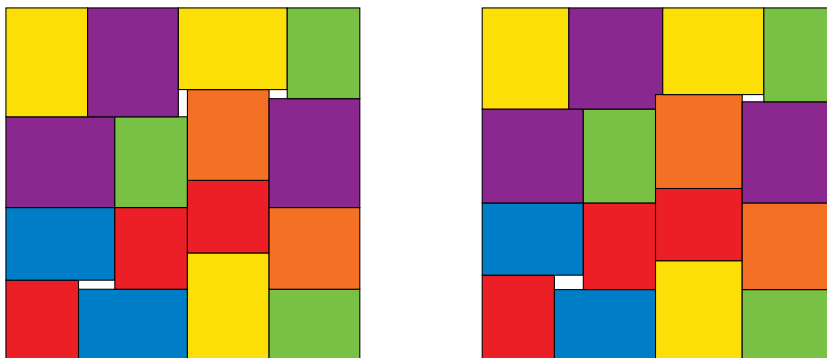
Corollary 4.7. 3 is a good dimension.

Proof. Take some representative P of the 21 unique universal packings found in Observation 4.3. Then by the proof above, P produces a packing of any increasing dimension tuple. Hence, P serves as a “recipe” for constructing a packing of *any* dimension tuple, whereby 3 is a good dimension. ■

5. EXPERIMENTS IN THE FOUR-DIMENSIONAL CASE

Let us examine the four-dimensional case. While the generalization of the problem to four dimensions is quite straightforward, the treatment of it is noticeably more difficult than that of lower dimensions. This is not only due to the greatly increased search space and subtle challenges in discarding symmetries. The real challenge arises from the fact that a RDTS must contain more than one dimension tuple. However, it took a long time to realize this and in order to get started, it was necessary to make a number of decisions, where the most appropriate choice was not obvious, simply because of the sparse knowledge of this problem.

Initially it seemed natural that examining any dimension tuple satisfying Hoffman’s inequality and with distinct elements would represent the “most difficult” case. This is the line of thought presented by Spiridonov in [Spi03, p. 6] and, after all, the immediate generalization of Hoffman’s focus in [Hof81, p. 215]. However, when attempting to count the number of four-dimensional squares it became clear that the number depended on the concrete choice of dimension tuple (satisfying Hoffman’s inequality and with distinct elements). Figure 12 gives an example of a square without overlap for one of these dimension tuples, but with overlap for a different one. This sudden overlap is due to a reliance on a non-trivial overlap



(a) Square of (8, 9, 10, 12) without overlap. (b) Square of (10, 12, 13, 14) with overlap.

FIGURE 12. Example of the same square produced using two different dimension tuples. Using the second dimension tuple produces an overlap.

inequality. Hence, it seemed appropriate to choose to completely exclude a partial solution if there is a pair of hyperrectangles where it relies solely on non-trivial overlap inequality to prevent overlap. This posed a problem, since—as we will see shortly—a four-dimensional dimension tuple will always satisfy at least one non-trivial overlap inequality. Hence, it was no longer possible to search for universal packings using a concrete dimension tuple, and this complicated the search considerably. However, this choice was later challenged by the discovery of a universal packing relying solely on non-trivial overlap inequalities for some of its overlap comparisons. It turns out that some non-trivial inequalities must be satisfied when others are not. Figure 13 illustrates an effect of this phenomenon. Here there are no overlaps in either of the two squares, but for some pairs of hyperrectangle this is due to a gap along one dimension in the first square and a different dimension in the second square. This discovery is the true motivation behind introducing the notion of a RDTS and the elaborate machinery surrounding it.

Example 5.1 (RDTS for $n = 4$). In this case all overlap inequalities have arity 1, 2 or 3. By Proposition 3.26 we only have to consider the overlap inequalities with

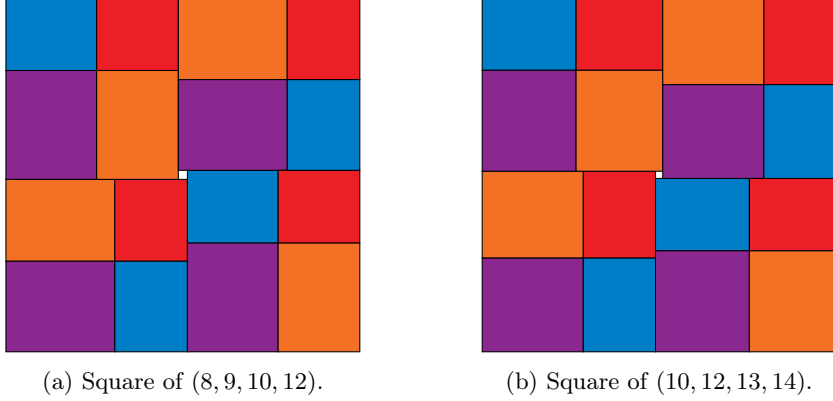


FIGURE 13. Example of square without overlap produced using two dimension tuples which satisfy different non-trivial overlap inequalities. This is surprising since the square relies on non-trivial overlap inequalities to prevent overlap.

arity 1 or 2. We would like to choose a dimension tuple (x_1, x_2, x_3, x_4) satisfying Hoffman's inequality and containing distinct elements. Of the overlap inequalities with arity 1, such a dimension tuple will only satisfy the trivial ones. If this is possible, then we only need to consider the overlap inequalities with arity 2. Suppose (A, B) is a non-trivial overlap inequality with arity 2. If $A \cap B \neq \emptyset$, then the inequality is really a non-trivial overlap inequality of arity 1 in disguise by Lemma 3.25. If $A \cap B = \emptyset$, then the only possibilities are

$$x_1 + x_4 \leq x_2 + x_3 \quad x_2 + x_3 \leq x_1 + x_4 \quad x_2 + x_4 \leq x_1 + x_3 \quad \text{and} \quad x_3 + x_4 \leq x_1 + x_2.$$

However, if we choose dimension tuples with distinct elements, then the last two will not be satisfied. Thus, we are left with

$$x_1 + x_4 \leq x_2 + x_3 \quad \text{and} \quad x_2 + x_3 \leq x_1 + x_4.$$

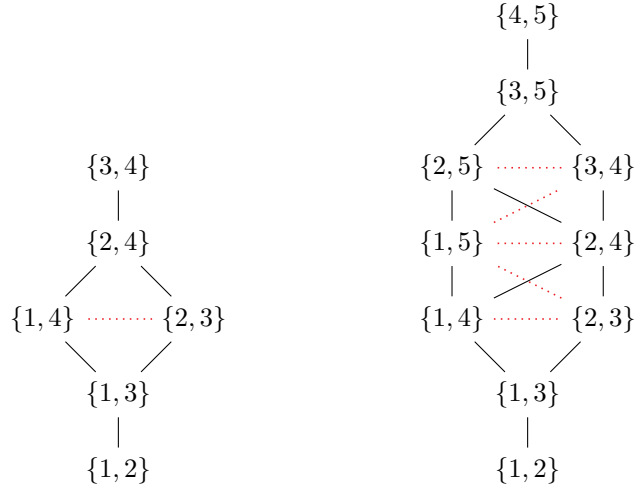
Observe that if one is not satisfied, then the other must be satisfied. However, it depends on the concrete choice of dimension tuple, whether only the first, only the second or both of them are satisfied. Observe that $d_1 = (8, 9, 10, 12)$ and $d_2 = (10, 12, 13, 14)$ both satisfy Hoffman's inequality and both contain distinct elements. Notice that d_1 does not satisfy the first overlap inequality, since $8 + 12 > 9 + 10$, while d_2 does not satisfy the second, since $12 + 13 > 10 + 14$. Hence $[d_1]$ and $[d_2]$ are the minimal elements and by Proposition 3.22

$$T = \{(8, 9, 10, 12), (10, 12, 13, 14)\}$$

is a RDTS for $n = 4$. Intuitively, we have covered every possible way that we can linearly order the sums of two distinct terms (without any two sums with different terms being equal). This is illustrated in Figure 14a.

Remark 5.2. Spiridonov considers the dimension tuple $d = (99, 100, 101, 102)$, because “the sizes differ by a very small amount” [Spi03, p. 6]. Notice that d satisfies both of the non-trivial overlap inequalities above as $99 + 102 = 100 + 101$. Hence, if Spiridonov were to find a Hoffman packing of d , it might not produce a packing of neither d_1 nor d_2 in the example above. In particular, it might not be a universal packing.

5.1. Utilizing approaches from the three-dimensional case. Let us begin by considering the dynamic programming approach.



(a) The four-dimensional case. (b) The five-dimensional case.

FIGURE 14. Hasse diagrams of the partial ordering of sums with two distinct terms. Dotted lines represent relations which can not be determined in general. Each set $\{i, j\}$ can be thought of as the sum $x_i + x_j$.

5.1.1. *Dynamic programming.* We have used the dynamic programming approach to construct all squares in the four-dimensional case using the same strategy as in the three-dimensional case, that is by first constructing all lines and then combining them into squares. Note that this time a square is not automatically specialized as it was in the three-dimensional case.

Observation 5.3. There are $4! = 24$ possible lines and if we specify one more dimension there are $24 \cdot 3^4 = 1944$. Using dynamic programming we have determined that there are $S = 51\,247\,458$ squares of the RDTS from Example 5.1 without overlaps and satisfying the Line criterion 3.5. Observe that our count is different from—and quite a bit lower—than the $77\,436\,138$ squares counted by Spiridonov [Spi03, p. 6] using the dimension tuple, which we questioned the choice of in Remark 5.2. We have reproduced his results for that choice of dimension tuple, but we still maintain that S is the appropriate number due to its independence of the choice of dimension tuple.

It takes a few minutes to generate all of the squares in the four-dimensional case and as we will see below, constructing all cubes does not seem to be practically possible for now. Let us attempt to estimate the number of cubes in the four-dimensional case and the computations needed to determine the exact number, if we construct them by stacking four squares. We only consider the consequences of the Line criterion 3.5, since it is difficult to predict overlap.

First, let us consider how many ways we can specialize the $4^2 = 16$ rectangles in a single square. There are two choices for each one, giving 2^{16} possible specializations. Hence, letting S be defined as in Observation 5.3 there are $2^{16}S$ ways to choose the first square. Next, we would like to stack a second specialized square on top of it. For simplicity and due to the inclusion of symmetries we suppose that there is a $3/4$ probability that the heights of two bricks on top of each other will be different, so we estimate around $(3/4)^{16}2^{16}S$ specialized squares to be compatible with the first choice. Hence, we estimate there to be around $(3/4)^{16}2^{32}S^2 = 3^{16}S^2$

half-cubes. At his point, the half-cubes are presumably so constrained by the Line criterion 3.5 that we consider an estimate of the number of compatible squares for the third square as too much guesswork. However, we can classify the half-cubes into 6^{16} groups depending on the $\binom{4}{2} = 6$ different ways to choose the brick heights in each of the 16 stacks of two bricks. Under the assumption that the half-cubes distribute evenly into these groups, we estimate that one can exhaust the search space by checking

$$(3^{16} S^2)^2 6^{-16} \approx 4.5 \cdot 10^{33}$$

different combinations of four squares. It is difficult to predict how many of these will have overlapping bricks. We have found $U = 1119514176$ different squares satisfying the Line criterion 3.5, if we do not check for overlaps while only $S/U \approx 4.58\%$ are in fact without overlapping bricks. Since there are more possibilities for overlaps in a cube, we suspect the actual number of cubes without overlapping bricks to be several orders of magnitude lower, but still quite large.

Next, let us consider the backtracking approach where we have been able to relatively quickly find a cube in the four-dimensional case.

5.1.2. *Backtracking.* We have tried out the procedures described in Table 2, but it has not been possible to find a solution within a reasonable time frame. However, each of the procedures finds a cube satisfying the Subgrid criterion 5.10 in around a minute. This gives us a basis for comparing the different procedures. Again, procedure D is a notable improvement over the naive procedure A requiring 84% fewer iterations to find a cube. The details can be found in Table 3. This improvement and the significance of the different criteria is illustrated in Figure 15. Notice that the detection of sharp corners (procedure B) gives the most significant improvement as opposed to checking the Subgrid criterion 5.10 (procedure C), while it was the other way around in the three-dimensional case.

TABLE 4. Number of iterations needed to find a cube satisfying the Subgrid criterion 5.10 in the four-dimensional case depending on the different test procedures.

Procedure	A	B	C	D
First cube	209 073 771	44 589 896	169 640 571	33 419 146

Using backtracking with procedure D for several days does not increase our hope of finding a solution. As shown in Figure 16 the approach appears to be stuck at around 86 hyperrectangles placed. Hence, these two approaches does—at least in their current form—not show much promise of finding a solution in the four-dimensional case within a reasonable time frame. However, before proceeding any further, we present the following observation, which we referred to earlier.

Observation 5.4. In the four-dimensional case using backtracking it is possible to generate a large number of cubes within a few hours. After doing so we obtained over 900 000 cubes and selected a sample of 900 of these. We have investigated whether a cube in this sample produces a valid subpacking of increasing dimension tuples not necessarily satisfying Hoffman’s inequality. It turns out that every cube from the sample produces a valid subpacking of any increasing dimension tuple (x_1, x_2, x_3, x_4) with integer values $1 \leq x_1 \leq x_2 \leq x_3 \leq x_4 \leq 100$.

5.2. **Counting the number of unique squares.** We can still determine the number of unique squares and as a by-product also provide some interesting insights into exploiting the wish to ignore symmetries. Using either backtracking or dynamic programming, one can obtain the number of squares including symmetries in a few

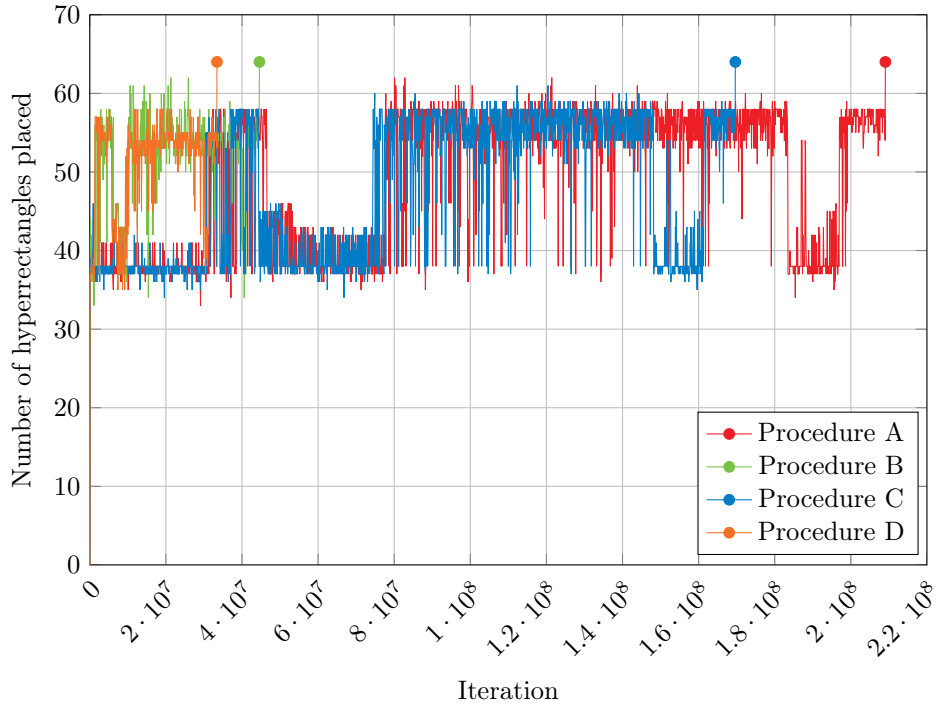


FIGURE 15. Plot of the number of hyperrectangles placed after a certain number of iterations in the four-dimensional case depending on the different test procedures. Here, we terminate as soon as a cube with $4^3 = 64$ bricks satisfying the Subgrid criterion 5.10 is found.

minutes. One would suspect this number—which we know to be $S = 51\,247\,458$ from Observation 5.3—to be divisible by $2! \cdot 2^2 = 8$, since each square has 8 symmetries. However, 8 does not divide S , so something does not add up. Let us investigate this discrepancy.

It turns out that since four is an even dimension, we can in fact come across squares where some symmetries coincide. A way to approach discarding symmetries is to try and mimic the concept of a “middle” brick, which we have in odd dimensions. We do so by considering the hyperrectangles around the “middle”. We will refer to this group as a *kernel*. A kernel consists of 4 rectangles in the case of a square in the four-dimensional case. Observe that if two squares are a symmetry of one another, then their kernels must be as well. Thus, suppose we compute all possible kernels and discard their symmetries. Then a square obtained by completing one kernel can never be a symmetry of the completion of a different kernel. This divides the search into smaller parts which can be solved independently. However, we are not completely done with the trickiness of symmetries. If a kernel has 8 distinct symmetries, i.e. none of them coincide, then any two different completions to a square can not be a symmetry of each other. However, for a kernel where some of its symmetries coincide, then any two different completions of it could potentially be a symmetry of one another. If this is the case we need to discard one of them.

Observation 5.5. We have found that there are 6 564 square kernels not violating the Line criterion 3.5 in the four-dimensional case. Notice that this number is not divisible by 8, so some of the kernels must have coinciding symmetries. There are 6 kernels, which have only 2 distinct symmetries and these can be seen in Figure 17.

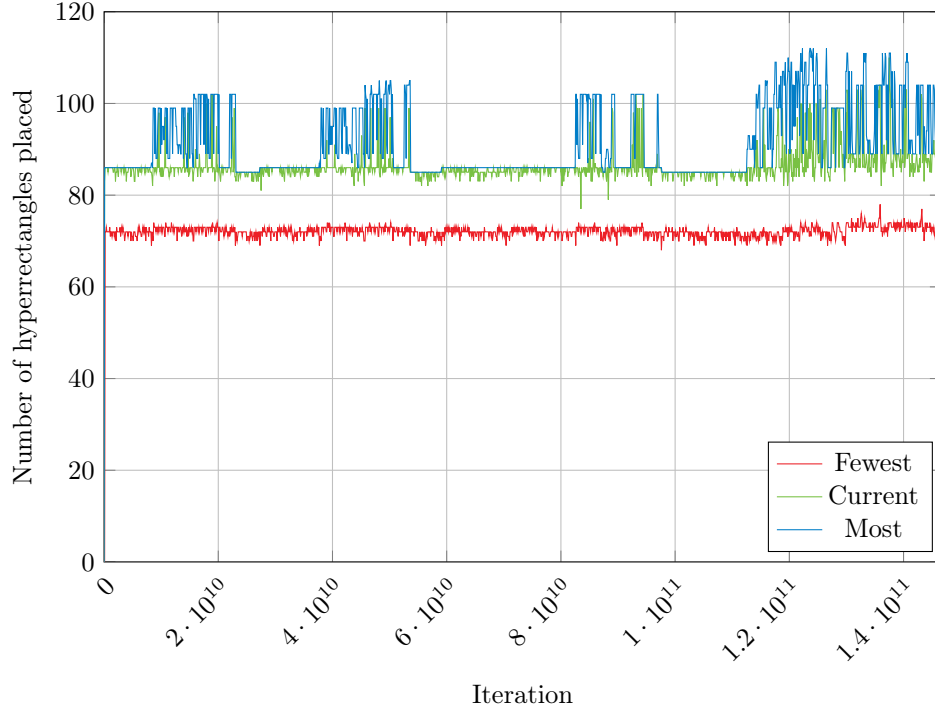


FIGURE 16. Plot of the number of hyperrectangles placed after a certain number of iterations in the four-dimensional case using procedure D. The number of hyperrectangles have been sampled every 10^8 iterations, each time noting the current number of hyperrectangles placed as well as the fewest and the most hyperrectangles having been placed since the previous sample was taken.

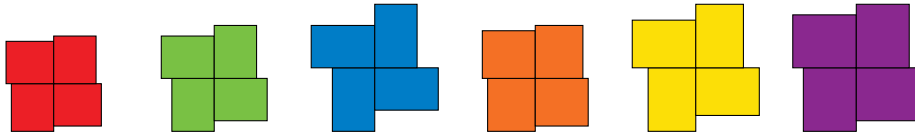


FIGURE 17. Square kernels in the four-dimensional case which have only 2 distinct symmetries.

Such a kernel consists of 4 of the same type of rectangle and the only distinct symmetry is obtained by a reflection. There are 18 kernels, which have only 4 distinct symmetries and these can be seen in Figure 18.

Such a kernel consists of two pairs of rectangles and the only distinct symmetries are obtained by a reflection and/or a rotation by $\pi/2$. Finally, there are 810 kernels with the ordinary 8 distinct symmetries. Hence, there are exactly 834 unique square kernels and this corresponds to the number above since

$$810 \cdot 8 + 18 \cdot 4 + 6 \cdot 2 = 6564.$$

Notice that we will have to discard symmetries after determining all completions of a kernel with coinciding symmetries. With this in mind, we have found that there are 6406310 unique squares satisfying the Line criterion 3.5 and without overlapping bricks for any dimension tuple found in Example 5.1.

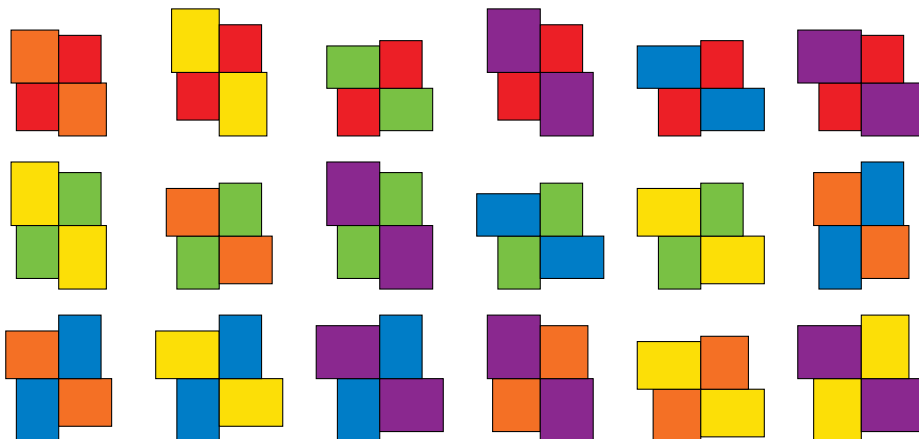


FIGURE 18. Square kernels in the four-dimensional case which have only 4 distinct symmetries.

5.3. Combining two-dimensional packings. It would seem that better branch pruning is needed to find universal packings in the four-dimensional case. One way of obtaining this is to find more criteria. However, they are hard to come by, especially when we do not have a four-dimensional universal packing to examine. Such a packing could help us to come up with conjectures and provide counterexamples. This sparked the idea to follow the constructive proof of Theorem 2.7 in order to obtain a four-dimensional packing.

Observation 5.6. We have combined two-dimensional packings into a four-dimensional one by following the proof of Theorem 2.7. As the two-dimensional packing we have used the one from Figure 2, which is the only unique universal packing for $n = 2$. Using a computer program, we have repeatedly used this solution to pack the groups of hyperrectangles in the proof of Theorem 2.7. This resulted in a four-dimensional universal packing, which is visualized in Figure 19.

This technique enables us to obtain several universal packings, because at each step of the proof, we have a choice of how we solve a group. Here we can use any of the two symmetries of the packing in Figure 2.

Remark 5.7. Strictly speaking, we never prove that combining Hoffman packings following the proof of Theorem 2.7 will actually result in a Hoffman packing. We leave this for others to formalize and prove thoroughly. We speculate that they might also be able to prove that such a combination preserves stability under permutations to some extent.

These universal packings behave extremely nice.

Observation 5.8. Notice that the two-dimensional packing in Figure 2 is stable under any permutation from S_2 . We have examined whether the universal packings constructed in Observation 5.6 are stable under permutations from S_4 . It turns out that they are stable under any permutation from S_4 . We speculate that this property is inherited through the proof of Theorem 2.7. It is dubious whether all four-dimensional universal packings exhibit this property. It is worth noting that this investigation also resulted in the universal packing from which Figure 13 originates and thus sparked the investigation into a RDTS.

5.4. The subgrid criterion. The four-dimensional universal packings constructed in Observation 5.6 provided the inspiration for how to generalize the last major piece

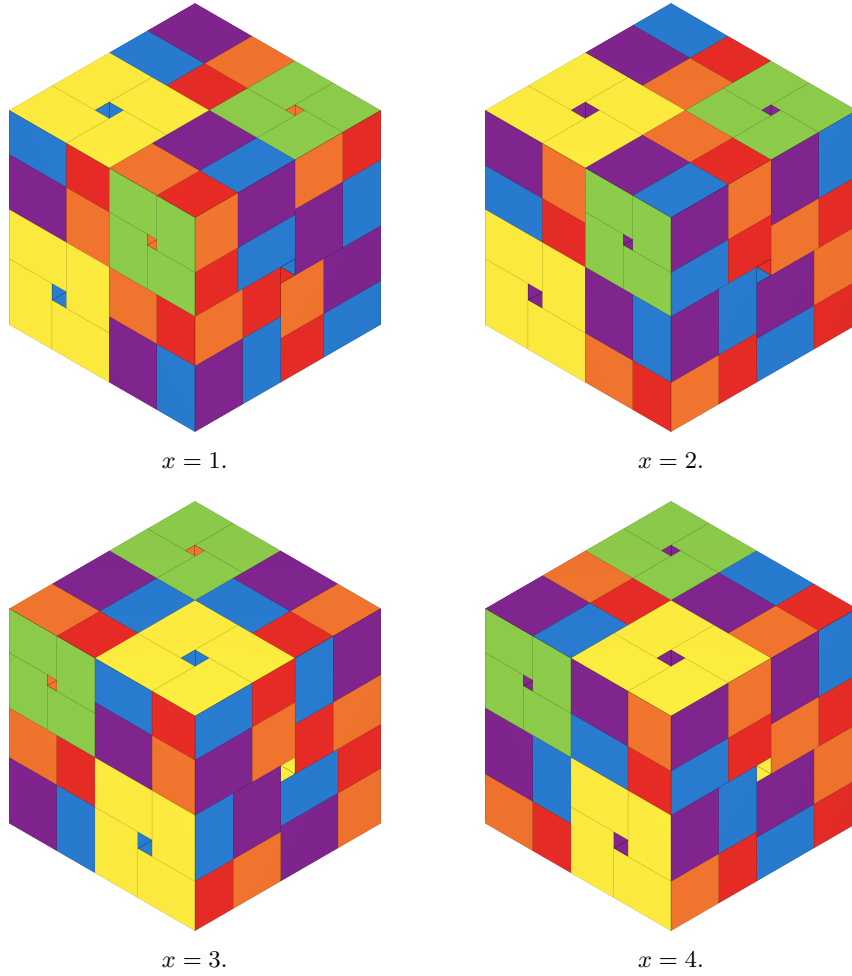


FIGURE 19. Visualization of the four cubes of a four-dimensional universal packing where the x -axis is fixed. The packing has been produced using the dimension tuple $(8, 9, 10, 12)$.

of knowledge provided by Hoffman in [Hof81, p. 221] about the three-dimensional problem. Let (x_1, x_2, x_3) be an increasing dimension tuple with distinct elements. Consider a square of it produced by a three-dimensional universal packing. Such a square contains $3^2 = 9$ rectangles, and Hoffman notes that these 9 rectangles must comprise exactly 3 x_1 -by- x_2 's, x_1 -by- x_3 's and x_2 -by- x_3 's. This is illustrated in Figure 3 where each type of rectangle has been given its own color. This observation has proven to be effective at pruning the search tree as seen in Figure 9 and Table 3. Hence, we have had a great interest in generalizing this observation to higher dimensions in the hope that it will be similarly effective there.

Observation 5.9. The following is the case for any of the universal packings constructed in Observation 5.6. Let (x_1, x_2, x_3, x_4) denote the dimension tuple with distinct elements, which has been used. Consider some cube of G_4 and note that it produces a subpacking containing $4^3 = 64$ bricks. These 64 bricks comprise exactly 16 x_1 -by- x_2 -by- x_3 's, x_1 -by- x_2 -by- x_4 's, x_1 -by- x_3 -by- x_4 's and x_2 -by- x_3 -by- x_4 's.

This provides enough information to put forward the following criterion.

Criterion 5.10 (Subgrid criterion). Suppose $n \geq 2$ and suppose $P: G_n \rightarrow S_n$. We say that P satisfies the *Subgrid criterion* if any $(n-1)$ -dimensional subgrid U of G_n satisfies the following: Suppose $d = (x_1, x_2, \dots, x_n)$ is an increasing dimension tuple with distinct elements and consider the subpacking produced by P restricted to U . This subpacking (consisting of n^{n-1} $(n-1)$ -dimensional hyperrectangles) contains precisely n^{n-2} hyperrectangles without an extent of x_i for all $i = 1, 2, \dots, n$.

Even better, we have been able to prove that this is a necessary condition for being a universal packing.

Proposition 5.11. Suppose $n \geq 2$ and suppose $P: G_n \rightarrow S_n$ is a universal packing. Then P satisfies the Subgrid criterion 5.10.

Proof. Note that P satisfies the Line criterion 3.5 by Proposition 3.8. Take some $(n-1)$ -dimensional subgrid U of G_n and consider the subpacking produced by P restricted to U . There are $(n-1)n^{n-2}$ grid lines in U , since there are n^{n-2} grid lines along each of the $n-1$ different dimensions. Take some i in $\{1, 2, \dots, n\}$. For each of these grid lines there must be precisely one hyperrectangle on it with an extent of x_i along the dimension of the grid line. Hence, there are at least $(n-1)n^{n-2}$ hyperrectangles in the subpacking with an extent of x_i along some dimension. Notice that the subpacking consists of n^{n-1} $(n-1)$ -dimensional hyperrectangles with a total number of $(n-1)n^{n-1}$ extends. Since i was chosen arbitrarily, it follows that there must be precisely $(n-1)n^{n-2}$ hyperrectangles in the subpacking with an extent of x_i along some dimension. Then the subpacking consists of precisely

$$n^{n-1} - (n-1)n^{n-2} = n^{n-2}$$

$(n-1)$ -dimensional hyperrectangles without an extent of x_i . Hence, P satisfies the Subgrid criterion 5.10. ■

Example 5.12. Suppose we have a packing produced by a universal packing. In the three-dimensional case we can conclude that there will be $3^{3-2} = 3$ of each of the 3 different types of rectangles in each square. In the four-dimensional case we can conclude that there must be $4^{4-2} = 16$ of each of the 4 different types of bricks in each cube.

In [Hof81] Hoffman puts great emphasis on the close relationship between inequalities and packing problems and the Subgrid criterion does also have an interesting interpretation as an inequality.

Corollary 5.13. Suppose $n \geq 2$ and suppose there exists a universal packing $P: G_n \rightarrow S_n$. Then for any dimension tuple $d = (x_1, x_2, \dots, x_n)$ satisfying Hoffman's inequality, we have that

$$n^{n-2} \left(\sum_{i=1}^n \frac{1}{x_i} \right) \left(\prod_{i=1}^n x_i \right) \leq \left(\sum_{i=1}^n x_i \right)^{n-1}.$$

Proof. Take some $(n-1)$ -dimensional subgrid U of G_n and consider the subpacking of d produced by P restricted to U . Pick some j in $\{1, 2, \dots, n\}$ and let us determine the combined hypervolume of the hyperrectangles without an extent of x_j . By the Subgrid criterion 5.10 there are n^{n-2} of these hyperrectangles yielding a combined hypervolume of

$$\frac{n^{n-2}}{x_j} \prod_{i=1}^n x_i.$$

Then the combined hypervolume of all of the hyperrectangles in the subpacking is

$$n^{n-2} \left(\sum_{i=1}^n \frac{1}{x_i} \right) \left(\prod_{i=1}^n x_i \right).$$

Notice that the surrounding $(n - 1)$ -dimensional hypercube has an extent of $x_1 + x_2 + \dots + x_n$ and hence a hypervolume of $(x_1 + x_2 + \dots + x_n)^{n-1}$. All of the $(n - 1)$ -dimensional hyperrectangles are pairwise non-overlapping and contained inside the surrounding $(n - 1)$ -dimensional hypercube. Hence, the inequality holds. ■

This is interesting, since if the above inequality does not hold in general, then it might be possible to rule out the existence of a Hoffman packing for certain dimension tuples and perhaps even show that a dimension is not good. However, it turns out that this inequality does in fact hold in general, since it follows quite easily [R17] from Maclaurin's inequality [Cve12, p. 188]. For $n = 4$, Maclaurin's inequality states that

$$(5) \quad \frac{x_1 + x_2 + x_3 + x_4}{4} \geq \sqrt{\frac{x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4}{6}}$$

$$(6) \quad \geq \sqrt[3]{\frac{x_1x_2x_3 + x_1x_2x_4 + x_1x_3x_4 + x_2x_3x_4}{4}}$$

$$(7) \quad \geq \sqrt[4]{x_1x_2x_3x_4}.$$

for any positive real numbers x_1, x_2, x_3 and x_4 . The inequality given by the left-hand side of (5) combined with (7) expresses the AM-GM inequality, while the inequality given by the left-hand side of (5) combined with (6) expresses the inequality treated above for $n = 4$. Inspired by Hoffman's focus on the close relationship between inequalities and packing problems let us see if we can find a connection between (5) and the four-dimensional problem. This inequality can be rearranged to

$$(8) \quad 16(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4) \leq 6(x_1 + x_2 + x_3 + x_4)^2$$

or equivalently

$$(9) \quad 8(x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4) \leq 3(x_1 + x_2 + x_3 + x_4)^2,$$

and the power of two on the right-hand side motivates us to look for a connection among the squares.

Observation 5.14. The following is the case for any of the universal packings constructed in Observation 5.6. Let (x_1, x_2, x_3, x_4) denote the dimension tuple with distinct elements, which has been used. Take any 3 pairwise "orthogonal" squares and consider the $3 \cdot 4^2 = 48$ rectangles in them. These 48 rectangles comprise exactly 8 x_1 -by- x_2 's, x_1 -by- x_3 's, x_1 -by- x_4 's, x_2 -by- x_3 's, x_2 -by- x_4 's and x_3 -by- x_4 's. The phenomenon can be observed in Figure 19 and all the squares of this packing can be found in Appendix B. Notice that this implies that "parallel" squares have the same number of each type of rectangle.

This corresponds precisely to the inequality (9). It has not been possible to prove that this is a necessary condition for being a universal packing. Neither has it been possible to come up with a counterexample. Even if this is not the case, a weaker claim might still hold, namely that there are 16 of each type of rectangle in any six pairwise "orthogonal" squares. This corresponds to the inequality (8).

In order to come up with a counterexample or strengthen the above two hypotheses, we need another way of constructing a four-dimensional universal packing. Since the ordinary backtracking and dynamic programming approaches does not show much promise, we have also tried an alternative approach.

Observation 5.15. Starting from one of the universal packings constructed in Observation 5.6, we removed a number of hyperrectangles and then attempted to use backtracking to complete this partial solution. We could remove up to 94 hyperrectangles without completely derailing the backtracking algorithm. Without

terminating after two days it had found $C = 435\,122\,478$ universal packings. A few random samples of these did not provide a counterexample to any of the two hypotheses above. These solutions appear to have the same overall structure as the universal packings from Observation 5.6, but with some of the hyperrectangles swapped around. This is illustrated in Figure 20. Since any unique four-dimensional universal packing has at most $4! \cdot 2^4 = 384$ non-coinciding symmetries by Proposition 3.4 there must be at least $\lfloor C/384 \rfloor = 1\,133\,131$ unique universal packings in the four-dimensional case.

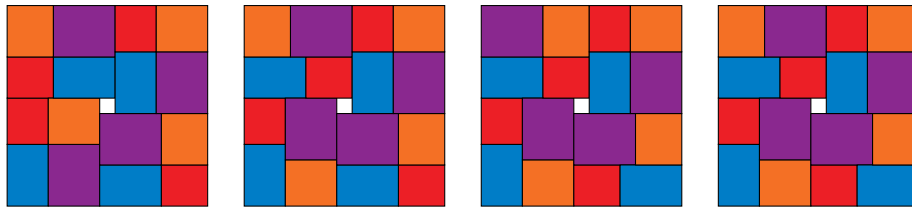


FIGURE 20. Example of how completing a partial solution in multiple ways might lead to a swapping of some of the hyperrectangles.

This observation indicates that some regions of the search space might have a particularly high density of solutions and also that there are a *lot* of four-dimensional unique universal packings compared to the three-dimensional case.

6. APPROACHING THE FIVE-DIMENSIONAL CASE

Due to time constraints our research into the five-dimensional case has been rather sparse. However, our formalization of the problem and the theoretical framework introduced does enable us to provide a few insights. We begin with a RDTS.

Example 6.1 (RDTS for $n = 5$). In this case all overlap inequalities have arity 1, 2, 3 or 4. By Proposition 3.26 we only have to consider the overlap inequalities with arity 1 or 2. We would like to choose a dimension tuple $(x_1, x_2, x_3, x_4, x_5)$ satisfying Hoffman’s inequality and containing distinct elements, since by doing so we can make sure that of the overlap inequalities with arity 1, such a dimension tuple will only satisfy the trivial ones. Then we only need to consider the overlap inequalities with arity 2. Next, we follow the same line of thought as in Example 5.1 where it resulted in a pair of non-trivial overlap inequalities, where at most one could not be satisfied. This resulted in an RDTS with two dimension tuples. Now, we end up with 5 pairs, namely

$$\begin{aligned} &(A, B), (B, A), (B, C), (C, B), \\ &(C, D), (D, C), (C, F), (C, F), \\ &(E, F), (F, E), \end{aligned}$$

where $A = \{1, 4\}$, $B = \{2, 3\}$, $C = \{1, 5\}$, $D = \{2, 4\}$, $E = \{2, 5\}$ and $F = \{3, 4\}$. Hence, any dimension tuple must always satisfy at least five non-trivial overlap inequalities. Intuitively, we are attempting to cover every possible way to linearly order the sums of two distinct terms (without any two sums with different terms being equal) and this is illustrated in Figure 14b. Then, one might suspect there to be $2^5 = 32$ minimal equivalence classes of dimension tuples, but this is not the case. Some of these overlap inequalities are in fact mutually dependent on one another in an intricate way as illustrated in Figure 21. By looking at this decision tree we

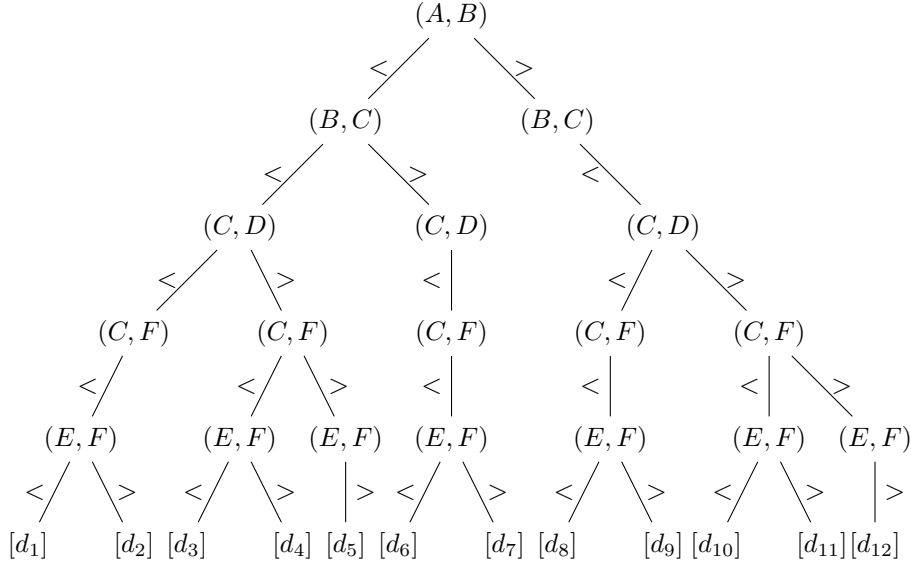


FIGURE 21. Decision tree showing how not satisfying certain overlap inequalities affect other overlap inequalities. It is quite helpful to compare this with Figure 14b.

observe that there are in reality only 12 minimal elements. For each of these we

have found a representative satisfying Hoffman’s inequality, namely

$$\begin{aligned} d_1 &= (28, 31, 34, 36, 38), & d_2 &= (22, 25, 26, 28, 30), & d_3 &= (26, 28, 32, 33, 36), \\ d_4 &= (20, 22, 24, 25, 28), & d_5 &= (18, 20, 21, 22, 26), & d_6 &= (22, 25, 27, 28, 29), \\ d_7 &= (24, 28, 29, 30, 32), & d_8 &= (22, 24, 26, 29, 30), & d_9 &= (26, 29, 30, 34, 36), \\ d_{10} &= (20, 21, 24, 26, 28), & d_{11} &= (24, 26, 28, 31, 34), & d_{12} &= (15, 16, 17, 19, 22). \end{aligned}$$

Hence, the set of these 12 dimension tuples constitutes a RDTS for $n = 5$.

In the four-dimensional case we immediately began counting the number of unique universal packings, since 4 is a good dimension by Theorem 2.7. However, we do not know whether 5 is a good dimension and whether a universal packing exists. Until then we might be better off trying to find a Hoffman packing of each of these 12 dimension tuples individually. Intuitively, they represent the 12 most difficult types of dimension tuples to solve.

Remark 6.2. What about the criteria found to be necessary properties of universal packings? Interestingly, each of the 12 dimension tuples above has properties similar to those of the dimension tuple in Lemma 3.2, whereby any Hoffman packing of it must necessarily satisfy the Line criterion 3.5. Then the Subgrid criterion 5.10 must hold as well. In fact, we suspect that in higher dimensions minimal equivalence classes of dimension tuples will always result in any Hoffman packing necessarily satisfying the Line criterion 3.5.

Next, we highlight a few investigations which we deem particularly interesting. They have not been pursued further due to time constraints.

6.1. Fundamentally different approaches. It would be interesting to explore the possibilities of making “local changes” to a pseudo-packing and use this to gradually reduce the number of overlapping hyperrectangles. One could begin by further exploring the results obtained in Observation 4.4.

It would also be interesting to explore the possibility of reducing a higher dimensional packing to a packing in lower dimensions. This would enable us to create a five-dimensional packing via a reduction of a six-dimensional one obtained via Theorem 2.7. We have made a few failed attempts at reducing a four-dimensional packing to a three-dimensional one. However, others might have better luck.

6.2. Ties to Maclaurin’s inequality. There are a number of ways to further explore the results obtained in Observation 5.14. First of all, it would be interesting to see if imposing such a criterion significantly speeds up the search process in the four-dimensional case. It is unclear whether this might be the case, since the Subgrid criterion 5.10 has proven quite effective at pruning the search tree in the three-dimensional case (as seen in Figure 9), but less effective in the four-dimensional case (as seen in Figure 15).

Note that it is still relevant to look for four-dimensional universal packings through other approaches in order to explore whether such a criterion is a necessary condition for being a universal packing.

We suspect that it might also be possible to put forward similar criteria for the five-dimensional case by looking at Maclaurin’s inequality for $n = 5$. A way to better justify imposing such rather speculative constraints would be to construct a six-dimensional universal packing by combining a two- and a three-dimensional universal packing and then examining whether it exhibits similar criteria for $n = 6$.

6.3. Formulation as a constraint satisfaction problem. During the course of this project, it has become increasingly clear that it is most natural to formulate Hoffman’s multidimensional packing problem as a constraint satisfaction problem (CSP). This class of problems has been subject to intense research in both artificial intelligence and operations research and one will thus be able to draw upon the advances in these fields. A CSP as described in [BPS99] and [Bar98] consists of a list of variables, a list of their respective value domains and a set of constraints over the variables. Then a solution is a choice of a value for each variable such that all constraints are satisfied. When solving a CSP we can look for one or all solutions and this corresponds perfectly to the types of questions raised about our packing problem.

The most common search algorithm for solving CSPs is backtracking, and here there are several ways to tune it, including the order in which variables are assigned values as well as the order in which compatible values are chosen.

For instance, one could employ a dynamic variable ordering in which the choice of the next variable to be assigned a value depends on the current state of the search. It is a common heuristic to select the variable with the fewest remaining compatible values. If the current partial solution can be completed to a solution, then every remaining variable must be assigned a value. The variable with the smallest domain is likely to be the most difficult to find a value for, since assigning values to other variables first may further reduce the number of compatible values.

Next, the order in which the values are tried out can also be adjusted. A different value ordering will rearrange the branches springing from each node of the search tree. This is an advantage if it results in searching a branch leading to a solution earlier than branches leading to dead ends. For instance, if a CSP has a solution and if a correct value is chosen for each variable, then a solution can be found without any backtracking.

Hence, formulating the problem as a CSP might enable us to prune the search tree thoroughly and let us experiment with finding the best variable ordering and value ordering. In retrospect the approaches tried out in this project may actually be thought of as immature variants of such a backtracking algorithm.

The backtracking approach presented earlier places one hyperrectangle at a time. However, a CSP solver might only specify some of the extends of a hyperrectangle and wait with the rest until later. It will also have more opportunities to experiment with variable ordering, whereas we could only place hyperrectangles on top of those already placed.

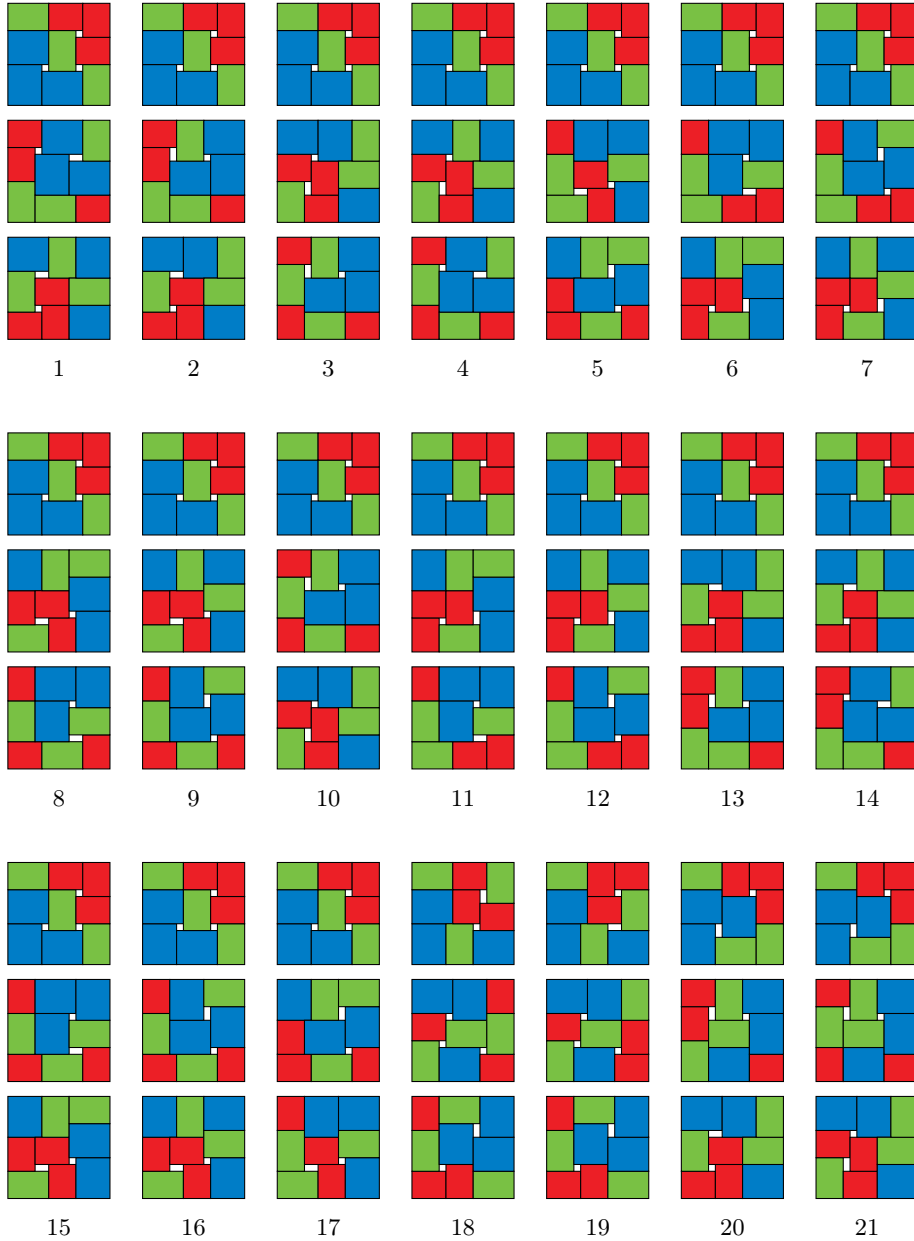
The dynamic programming approach can also be analyzed within this framework. It is really a search for gradually larger groups of mutually compatible assignments of values. Take for instance the case of constructing a cube by stacking squares. Here the constraints only related to the assigned square will always be satisfied. However, this analysis also reveals a weakness of the approach. Not only does it need a large amount of memory to store the mutually comparable values (e.g. all squares), it is also very questionable whether this variable ordering is the most effective. In particular, this consideration shows why it might not be helpful to combine for instance orthogonal squares. One would assign values to many variables with untouched domains, while possibly ignoring variables with very few compatible values left.

We speculate that a formulation as a CSP might be well-suited for searching for a four-dimensional universal packing and perhaps even attempt to determine the number of unique cubes in the four-dimensional case, where one should also be able to parallelize the search by dividing the problem into smaller parts using kernels.

With assistance from Trine K. Boomsma, we approached the packing problem using mixed-integer programming. This approach turned out to be ill-suited, since the search algorithms spent a lot of time looking for non-optimal solutions, while we are really only interested in optimal integer solutions. However, exploring this field made it more clear that approaching the problem using constraint satisfaction is well worth a try. We suspect that existing general purpose CSP solvers (commercial as well as free ones) are not flexible enough to take full advantage of the domain-specific knowledge of this problem. In particular, we fear that solving simultaneously for multiple dimension tuples (such as those in a RDTS) might not be possible. Also, we fear that constraints imposed by results obtained by investigations into the links to Maclaurin's inequality might be difficult to express. Hence, a custom implementation of a CSP solver might be worth looking into.

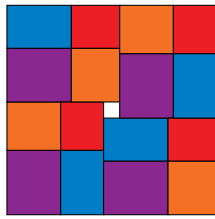
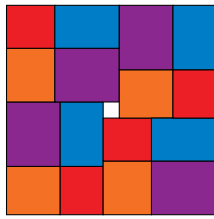
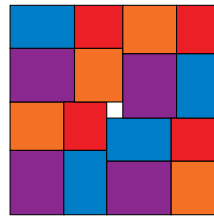
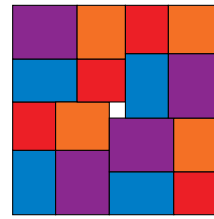
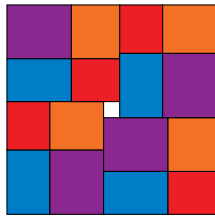
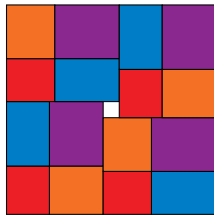
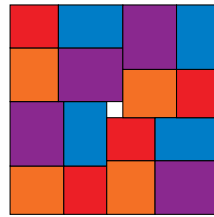
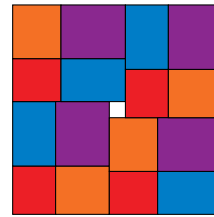
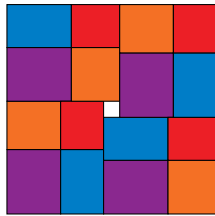
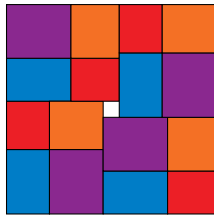
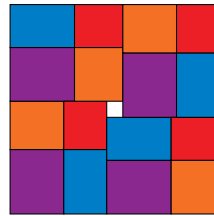
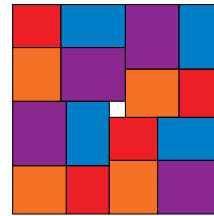
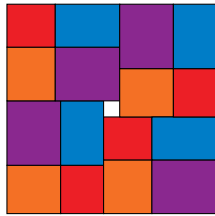
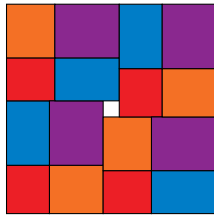
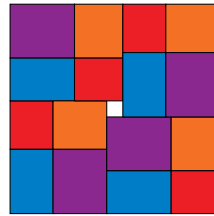
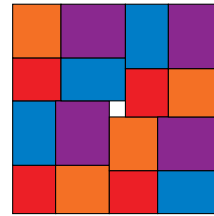
APPENDIX A. THREE-DIMENSIONAL UNIVERSAL PACKINGS

Here we present all of the 21 three-dimensional unique universal packings. They have been produced using the dimension tuple $(4, 5, 6)$. The 4-by-5, 4-by-6 and 5-by-6 rectangles are colored red, green and blue, respectively.

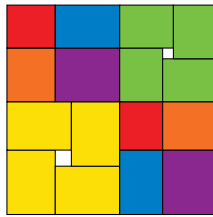
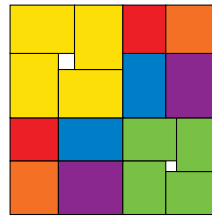
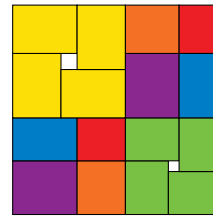
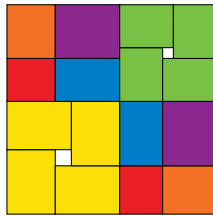
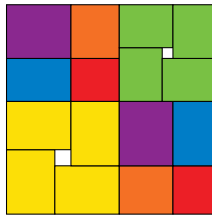
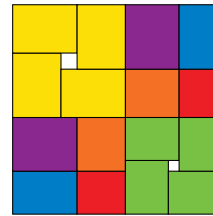
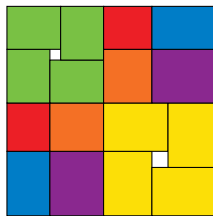
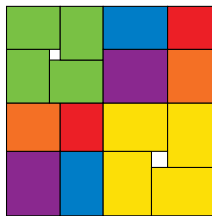
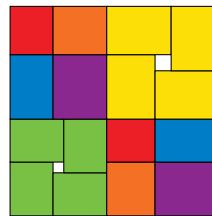
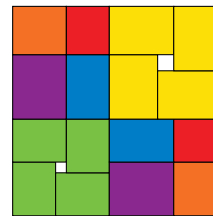
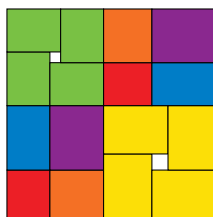
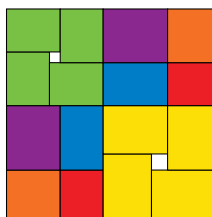
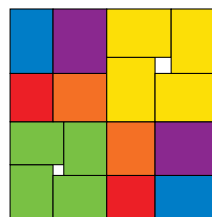
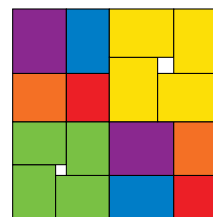


APPENDIX B. FOUR-DIMENSIONAL UNIVERSAL PACKING

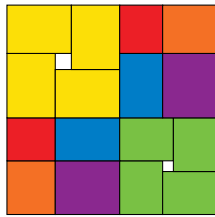
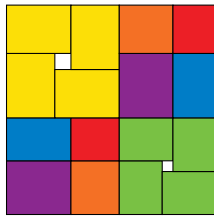
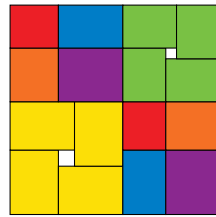
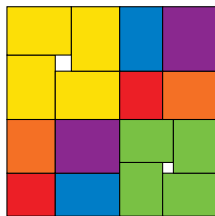
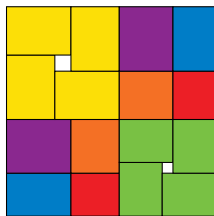
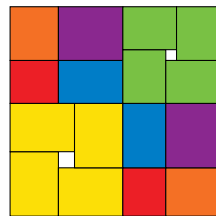
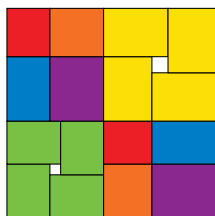
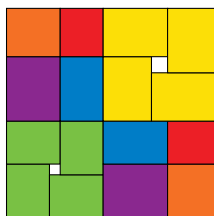
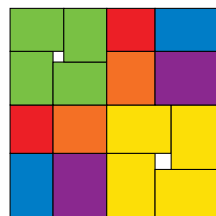
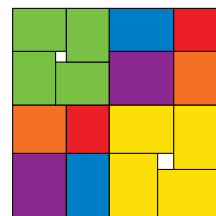
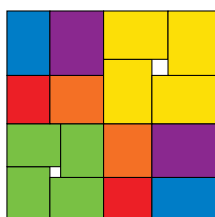
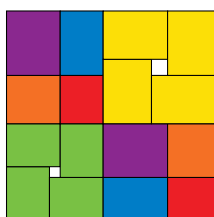
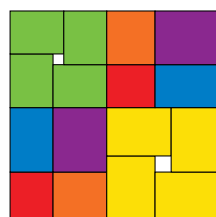
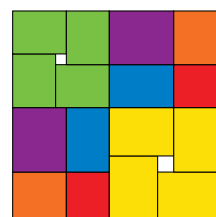
Here we present all of the squares of a four-dimensional universal packing. The packing has been produced using the dimension tuple $(8, 9, 10, 12)$. The 8-by-9, 8-by-10, 8-by-12, 9-by-10, 9-by-12 and 10-by-12 rectangles are colored red, green, blue, orange, yellow and violet, respectively.

 $x = 1$ and $y = 1$. $x = 1$ and $y = 2$. $x = 1$ and $y = 3$. $x = 1$ and $y = 4$. $x = 2$ and $y = 1$. $x = 2$ and $y = 2$. $x = 2$ and $y = 3$. $x = 2$ and $y = 4$. $x = 3$ and $y = 1$. $x = 3$ and $y = 2$. $x = 3$ and $y = 3$. $x = 3$ and $y = 4$. $x = 4$ and $y = 1$. $x = 4$ and $y = 2$. $x = 4$ and $y = 3$. $x = 4$ and $y = 4$.

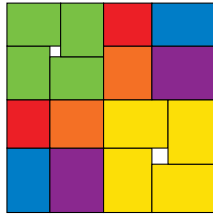
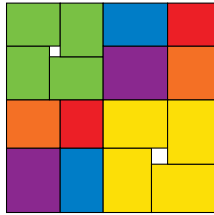
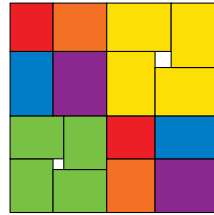
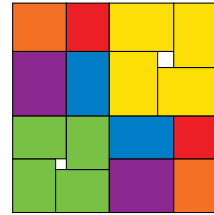
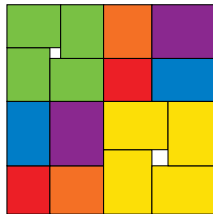
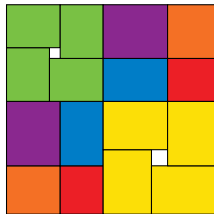
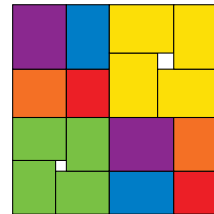
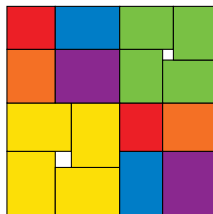
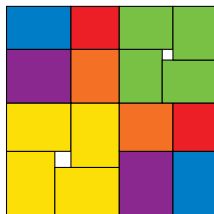
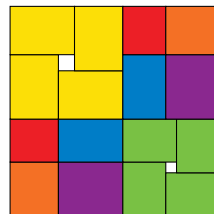
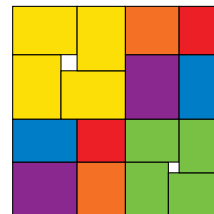
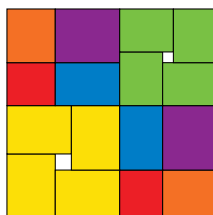
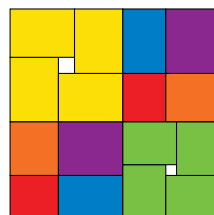
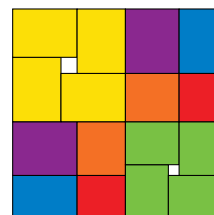
Squares where the x - and y -axes are fixed.

 $x = 1$ and $z = 1$. $x = 1$ and $z = 2$. $x = 1$ and $z = 3$. $x = 1$ and $z = 4$. $x = 2$ and $z = 1$. $x = 2$ and $z = 2$. $x = 2$ and $z = 3$. $x = 2$ and $z = 4$. $x = 3$ and $z = 1$. $x = 3$ and $z = 2$. $x = 3$ and $z = 3$. $x = 3$ and $z = 4$. $x = 4$ and $z = 1$. $x = 4$ and $z = 2$. $x = 4$ and $z = 3$. $x = 4$ and $z = 4$.

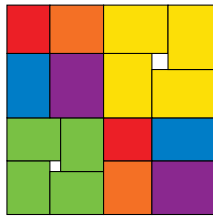
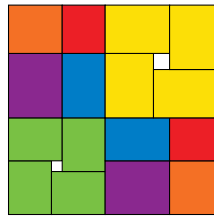
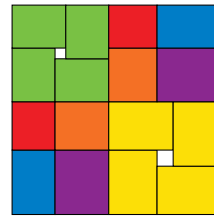
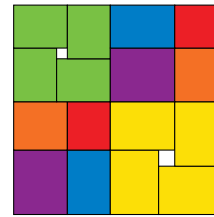
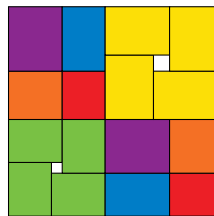
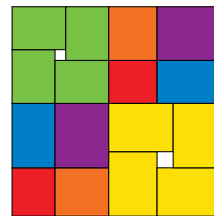
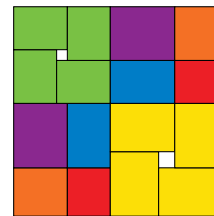
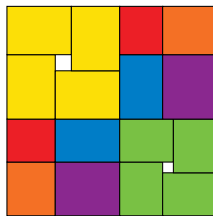
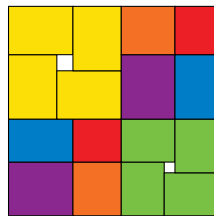
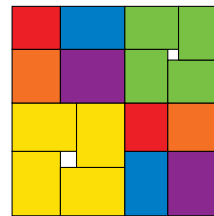
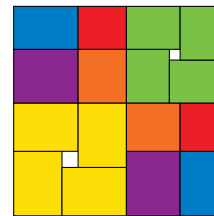
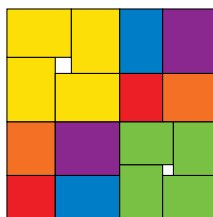
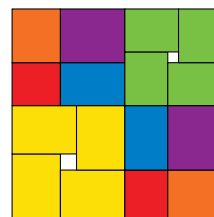
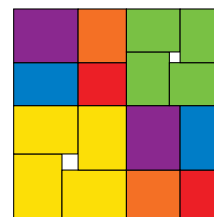
Squares where the x - and z -axes are fixed.

 $x = 1$ and $w = 1$. $x = 1$ and $w = 2$. $x = 1$ and $w = 3$. $x = 1$ and $w = 4$. $x = 2$ and $w = 1$. $x = 2$ and $w = 2$. $x = 2$ and $w = 3$. $x = 2$ and $w = 4$. $x = 3$ and $w = 1$. $x = 3$ and $w = 2$. $x = 3$ and $w = 3$. $x = 3$ and $w = 4$. $x = 4$ and $w = 1$. $x = 4$ and $w = 2$. $x = 4$ and $w = 3$. $x = 4$ and $w = 4$.

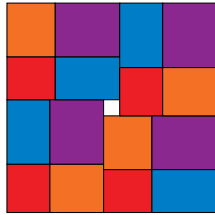
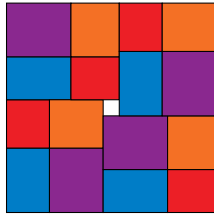
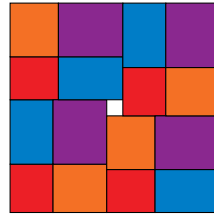
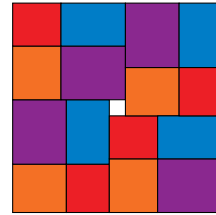
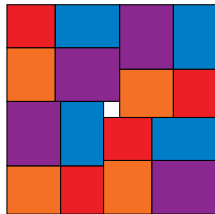
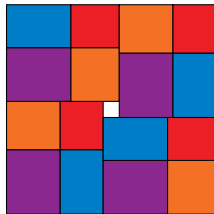
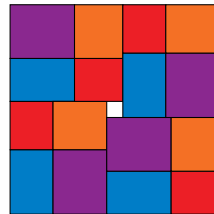
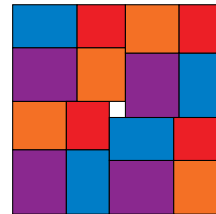
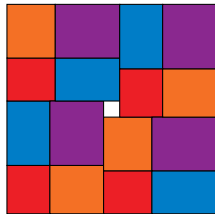
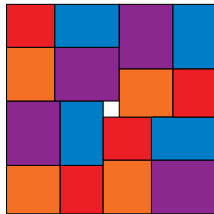
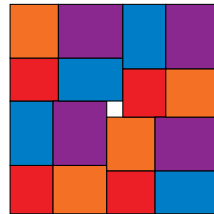
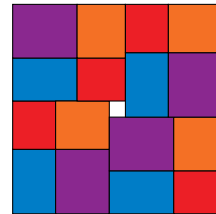
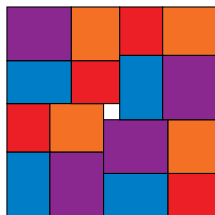
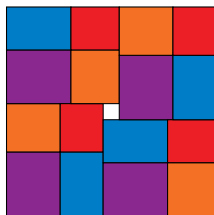
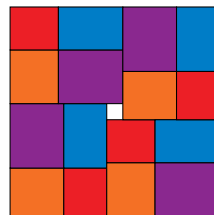
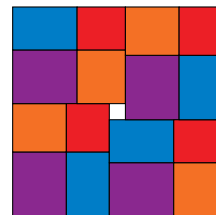
Squares where the x - and w -axes are fixed.

 $y = 1$ and $z = 1$. $y = 1$ and $z = 2$. $y = 1$ and $z = 3$. $y = 1$ and $z = 4$. $y = 2$ and $z = 1$. $y = 2$ and $z = 2$. $y = 2$ and $z = 3$. $y = 2$ and $z = 4$. $y = 3$ and $z = 1$. $y = 3$ and $z = 2$. $y = 3$ and $z = 3$. $y = 3$ and $z = 4$. $y = 4$ and $z = 1$. $y = 4$ and $z = 2$. $y = 4$ and $z = 3$. $y = 4$ and $z = 4$.

Squares where the y - and z -axes are fixed.

 $y = 1$ and $w = 1$. $y = 1$ and $w = 2$. $y = 1$ and $w = 3$. $y = 1$ and $w = 4$. $y = 2$ and $w = 1$. $y = 2$ and $w = 2$. $y = 2$ and $w = 3$. $y = 2$ and $w = 4$. $y = 3$ and $w = 1$. $y = 3$ and $w = 2$. $y = 3$ and $w = 3$. $y = 3$ and $w = 4$. $y = 4$ and $w = 1$. $y = 4$ and $w = 2$. $y = 4$ and $w = 3$. $y = 4$ and $w = 4$.

Squares where the y - and w -axes are fixed.

 $z = 1$ and $w = 1$. $z = 1$ and $w = 2$. $z = 1$ and $w = 3$. $z = 1$ and $w = 4$. $z = 2$ and $w = 1$. $z = 2$ and $w = 2$. $z = 2$ and $w = 3$. $z = 2$ and $w = 4$. $z = 3$ and $w = 1$. $z = 3$ and $w = 2$. $z = 3$ and $w = 3$. $z = 3$ and $w = 4$. $z = 4$ and $w = 1$. $z = 4$ and $w = 2$. $z = 4$ and $w = 3$. $z = 4$ and $w = 4$.

Squares where the z - and w -axes are fixed.

REFERENCES

- [Bar98] Roman Barták. Constraint programming. <https://ktiml.mff.cuni.cz/~bartak/constraints/index.html>, 1998. [Accessed: January 10, 2018].
- [BCG04] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for Your Mathematical Plays*. A K Peters, 2004.
- [BPS99] Sally C. Brailsford, Chris N. Potts, and Barbara M. Smith. Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119(3):557 – 581, 1999.
- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. The MIT Press, 2009.
- [Cut99] Bill Cutler. Block-packing jambalaya. In E.R. Berlekamp and T. Rodgers, editors, *The Mathemagician and Pied Puzzler: A Collection in Tribute to Martin Gardner*, pages 169–225. Taylor & Francis, 1999.
- [Cve12] Zdravko Cvetkovski. Newton’s inequality, maclaurin’s inequality. In *Inequalities: Theorems, Techniques and Selected Problems*, pages 117–119. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [EJ17] Søren Eilers and Rune Johansen. *Introduction to experimental mathematics*. Cambridge University Press, 2017.
- [Hof81] D. G. Hoffman. Packing problems and inequalities. In David A. Klarner, editor, *The Mathematical Gardner*, pages 212–225. Wadsworth International, 1981.
- [Hof17] Dean G. Hoffman. Private communication, 2017.
- [Knu04] Donald Knuth. Perfect packing. <http://www.puzzlepalace.com/#/puzzles/200625/solution>, 2004. [Accessed: December 6, 2017].
- [Mil06] George Miller. Perfect packing. <http://www.puzzlepalace.com/#/puzzles/200625>, 2006. [Accessed: December 6, 2017].
- [R17] Martin R. Inequality $am/hm \leq (am/gm)^n$ with packing problem interpretation. Mathematics Stack Exchange (<https://math.stackexchange.com/q/2488264>), 2017. [Accessed: January 3, 2018].
- [Spi03] Alexey Spiridonov. Hoffman’s packing puzzle in 3 and 4 dimensions. <https://alexey.im/papers/hoffman-puzzle-for-conway.pdf>, 2003. [Accessed: December 16, 2017].

INDEX

- Arithmetic mean, 3
- Criterion
 - line, 16
 - no neighbour overlap, 17
 - subgrid, 38
- cut, 10
- cut distance, 10
- Dimension tuple, 5
 - increasing, 8
 - representative set (RDTS), 20
 - satisfying overlap inequality, 21
 - sum of, 5
- Geometric mean, 3
- Good dimension, 3
- Grid, 10
- Grid line, 11
 - completely stuffed, 12
 - stuffing on, 12
 - unique, 16
- Grid point, 10
 - associated hyperrectangle, 13
 - coordinate, 11
 - neighbouring, 17
- Hoffman packing, 13
 - equivalent, 13
 - unique, 13
- Hoffman's inequality, 9
- Hypercube, 5
 - surrounding, 5
- Hyperrectangle, 5
 - anchor point of, 5
 - centre of, 5
 - dimension tuple of, 5
 - extend of, 5
 - general, 5
 - hypervolume of, 5
 - non-overlapping, 5
 - overlapping, 5
- Kernel, 35
- Overlap comparison, 20
- Overlap inequality, 21
 - arity of, 21
 - satisfied by dimension tuple, 21
 - trivial, 21
- Packing, 5
 - equivalence of, 8
 - produced by Hoffman packing, 13
 - pseudo-, 13
 - rearrangement of, 8
 - rigid, 8
 - symmetry of, 8
- RDTS, *see* Dimension tuple
- Specialization, 24
- Subgrid, 24
- Subpacking, 24
- Universal packing, 13
 - distance between, 26
 - equivalent, 13
 - stable under permutation, 27
 - unique, 13