

# Fractal Analysis of the Subset-Sum Problem

Ruben Horn<sup>1</sup><sup>a</sup>, Daan van den Berg<sup>2,3</sup><sup>b</sup> and Pieter Adriaans<sup>4</sup><sup>c</sup>

<sup>1</sup>*Helmut-Schmidt-University, Hamburg, Germany, U.K.*

<sup>2</sup>*Department of Computer Science, University of Amsterdam, Netherlands*

<sup>3</sup>*Department of Computer Science, VU Amsterdam, Netherlands*

<sup>4</sup>*Institute for Logic, Language, and Computation, University of Amsterdam, Netherlands*

**Keywords:** Subset-Sum Problem, Number Partitioning Problem, Fractal Analysis, Hilbert's Hotel, NP-Hard.

**Abstract:** It is known that the hardness of the (two-way) number partitioning problem (NPP) variant of the subset-sum problem (SSP) depends on the number and distribution of bits in the set of numbers, but beyond this, it is relatively unexplained for the SSP itself. Thus, we look at the solution space of various problem instances of the SSP using fractal analysis. Two methods to determine the dimension are used. Plotting the fractal dimension over the range and distributions of informational bits, we find that it is correlated with this linear model and also moderately correlated to the hardness of the NPP. This suggests that fractal analysis might be a useful tool in understanding the complexity of combinatorial problems and, we believe, may help further understand the hardness in NP. Finally, we introduce a thought experiment derived from the famous Hilbert's hotel, which we call Hilbert's hotel with elevators, to intuitively illustrate how the complexity of the solutions space and the computational hardness may relate across combinatorial problems.

## 1 INTRODUCTION AND RELATED WORK

The subset-sum problem (SSP) is a particularly interesting problem within the class of NP-complete (Garey and Johnson, 1990) problems because it is actually quite simple, both in terms of its definition and in terms of the computational hardness of many instances, which consist of a set of integers  $S \subset \mathbb{N}^+$  and a target value  $t \in \mathbb{N}$ . A subset  $A \subseteq S$  is a solution to this instance if  $\sum A = t$ . If  $t = \lceil \frac{1}{2} \sum S \rceil$ , this is known as the (two-way) number partitioning problem (NPP), which is also NP-complete (Karp, 1972).

Over the years, there have been several insights into what makes instances of this problem easy or hard. This is also of particular interest to the cryptography community, where the SSP has been the basis for an early asymmetric cryptosystem (Merkle and Hellman, 1978; Sharma et al., 2011).


**Lemma 1.** *Superincreasing SSP instances are trivial.*


*Proof.* A set  $S$  is superincreasing if all the integers it


contains are larger than the sum of all smaller ones contained in the set ( $\forall s \in S : s > \sum \{n | n < s\} \cap S$ ). Thus, the best possible subset for any  $t$  is either  $\min_{s \geq t} \{s\}$  or  $\min_{s \geq t} \{n | n < s\} \cap S$ , which are both trivial to find.  $\square$

Beyond superincreasing instances, the hardness of the problem for a pseudo-polynomial time algorithm depends on the magnitude of the integers in the instance (Kleinberg and Tardos, 2005, chapter 8.8). Korf (1998), Mertens (2003) and Hayes (2002) illustrated that the hardness of the NPP specifically is influenced by the ratio  $m/n$  of the average number of bits  $m$  required to represent each integer over the cardinality  $n$  of the set  $S$ . For the NPP, the relation of the number of solutions (perfect partitions) depends on this ratio  $m/n$ . If  $n$  is sufficiently large, the SSP becomes easy for at least some values of  $t$  by the sheer frequency of solutions, since  $\lim_{n \rightarrow \infty} n \times (2^m - 1) - 2^n < 0$ .<sup>1</sup> But even for the same value of the ratio  $m/n$ , some instances may differ in computational hardness, that is to say will be solved after fewer or more computational steps than others. This is not only attributable to the variation in the frequency of optimal partitions, but also depends on the concrete integers that make up the set.

<sup>1</sup>For very small  $m/n$ , multisets are unavoidable due to the limited number of different integers with  $m$  bits.

<sup>a</sup> <https://orcid.org/0000-0001-6643-5582>

<sup>b</sup> <https://orcid.org/0000-0001-5060-3342>

<sup>c</sup> <https://orcid.org/0000-0002-8473-7856>

Van Den Berg and Adriaans (2021) found that the distribution of informational bits over the integers in such random instances of the NPP influences the experimental hardness, which is determined for each instance by the number of recursions required to solve it using a depth first branch and bound (BB) algorithm. This algorithm generates candidate subsets from the list of integers sorted in decreasing order. Compared to the complete greedy and complete Karmarkar-Karp BB algorithms for the NPP by Korf (1998), which construct candidate subsets (leaf nodes) according to the respective heuristic from left to right, in the algorithm by Van den Berg and Adriaans each node is a candidate subset following a greedy heuristic and thus, the total search tree is much smaller.

In order to generate instances with varying distributions of informational bits across the integers, Van den Berg and Adriaans introduce the notion of strict templates, which determine the exact number of bits for every integer in the instance. They use seven different templates with 78 bits each over 12 integers to generate instances of the NPP, which consequently all have on average 6.5 bits per integer and so  $m/n \approx 0.54$ . Three templates are *eccentric*, meaning that their derivative has values greater than 1. Instances generated from such templates contain mostly small values and few relatively large ones. The *non-eccentric* ones range from the linearly increasing template  $\{1, 2, \dots, n\}$  to the (almost) uniform or *flat* template. From each of these templates, ten instances are randomly generated. For example, given a template  $\{3, 4, 5, 6, \dots\}$ , the first integer of a corresponding instance with exactly  $m_1 = 3$  bits may have any value between  $2^{m_1-1} = 4$  and  $2^{m_1} - 1 = 7$ . While instances derived from eccentric templates always have the same hardness, those generated from non-eccentric templates can be both easier or harder.

This experiment was replicated by Sazhinov et al. (2023) using 105 bits over 14 integers. They furthermore find that the different templates for generating instances also affect the performance of heuristic algorithms for the NPP.

Since both papers pertain to the NPP variant of the SSP, the parameter  $t$  is not covered. For every set  $S$ , there are some obvious values for  $t$ , which will result in trivial instances  $(S, t)$  that can be solved without backtracking. Beyond this, however, the solution landscape of the SSP is more opaque. If an instance is very eccentric, the histogram of the solution frequency for every value of  $t$  will have two large clusters on either side of the spectrum because all possible subset sum values are either very small or very large. For (almost) flat templates, the opposite is the case. Subset sum values around  $\lfloor \frac{1}{2} \sum S \rfloor$  are more frequent than

very small or very large ones. The picture is less clear between these two extremes, yet some patterns can be expected to emerge in the histogram of subset sum frequencies: Given some subset  $A \subset S$  of relatively small numbers, the histogram of all possible subsets of  $A$  will *occur* multiple times in the histogram for  $S$  at  $\{\sum B | B \subseteq S \setminus A\}$ . Examples for these three cases are visualized in Figure 1. Our hypothesis is that the SSP shows fractal properties, which may be correlated to its hardness. Consequently, we will use the fractal dimension to measure the statistical self-similarity in these histograms to describe the unpredictability and density of the solution space and thus its complexity (cf. Falconer, 2013, chapter 3).

In the following Section 2, we outline the methods for our two experiments, which includes generating and analyzing a dataset for each and describe the results in the subsequent Section 3. Beyond these experiments, we make a case for the importance of the complexity of the solution space for the computational hardness by illustrating it with a new thought experiment inspired by Hilbert's hotel (Ewald and Sieg, 2013) in Section 4. We conclude our work with a discussion of the implications and future work in Section 5 and 6.

All experiments were implemented in Python 3.9 and run on a dual-CPU compute node (72 cores at 2.4 GHz, 256 GB DDR4 memory) of the cluster HSUPER. A replication package including the dataset is available in an online repository (Horn et al., 2024b).

## 2 METHODS

For our experiment, we leverage the template approach by Van den Berg and Adriaans (2021) in order to generate random *proto-instances* (just the sets  $S$  without  $t$ ) of the SSP. The templates used for this result in different distributions of the informational bits across 15 integers. For each proto-instance the information distribution is characterized by the linear regression slope  $\beta$ , which is the logarithm of the sorted integers:

$$\log_2 s_i \approx \alpha + \beta i \text{ for } s_i \in S \quad (1)$$

The number of total bits  $\sum_i m_i$ , with  $m_i$  the number of bits for the  $i^{\text{th}}$  integer, is  $\frac{1}{2}n(n+1)$  where  $n = |S|$  as determined by the number of bits in the linearly increasing template ( $\beta = 1$ ), in which the number of bits for each integer corresponds to its index in the template. For  $n = 15$  the total number of bits is thus  $\sum_i m_i = 120$ .

**Lemma 2.** *The linearly increasing template ( $\beta = 1$ ) and flat template ( $\beta = 0$ ) are only possible for the same  $\sum_i m_i$  if  $n$  is odd.*

*Proof.* Otherwise,  $\frac{\sum_i m_i}{n} = \frac{1}{2n}n(n+1) = \frac{n+1}{2} \notin \mathbb{N}$ .  $\square$

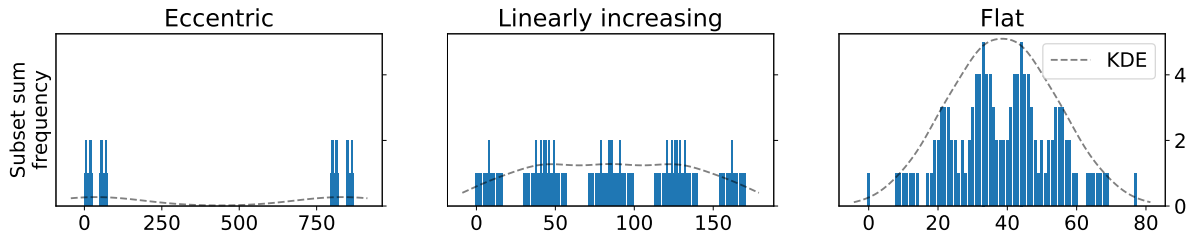


Figure 1: Histograms of the subset sum frequency and Gaussian kernel density estimation (KDE) for the eccentric set  $\{1, 2, 3, 5, 14, 50, 795\}$ , the linearly increasing set  $\{1, 2, 5, 8, 30, 41, 83\}$  and the flat set  $\{8, 9, 10, 11, 12, 13, 14\}$ . All three instances have 28 bits over 7 integers. The range of the horizontal axis is determined by the sum of each set.

To cover the spectrum of values for  $\beta$  as widely as possible, including both non-eccentric and eccentric templates with  $\beta \leq 1$  and  $\beta > 1$  respectively, three approaches are used to come up with templates from which random instances are generated. For the non-eccentric templates, the values for  $\beta$  should cover the full range  $[0, 1]$ . For eccentric ones, we chose a ramp function, as it is a simple method of realizing non-linear distributions which can still be distinguished in our dataset using  $\beta$  in Equation (1). The three methods used for generating templates are described below:

- *Non-eccentric templates* are generated from Equation (2) by distributing the number of bits as linearly as possible following the desired slope  $\beta \in (0; 1]$  with uniform distribution of the remaining bits ( $T_{\beta=0}$ ).

$$(\lceil i\beta \rceil | i \in \{1..n\}) + T_{\beta=0} \quad (2)$$

This approach predominantly generates templates with a low value for  $\beta$  closer to 0.

- *More non-eccentric templates* are generated by integrating all possible binary sequences  $\{0, 1\}^n$ . If the remaining bits cannot be evenly distributed over the integers, the template is discarded. Sampling with uniform spacing from all feasible templates generated using this approach predominantly yields templates with  $\beta$  closer to 1.
- *Eccentric templates* are generated from Equation (3) by concatenating a shorter flat template of 1s and a linearly increasing template ( $T_{\beta \geq 1}$ ) of length  $i \in [2..n)$  using all remaining bits.

$$((1,)^{n-i}, T_{\beta \geq 1}(i)) \quad (3)$$

This inevitably results in proto-instances containing multisets. However, this is unavoidable if the ratio  $m/n$  should be the same between all templates.

Admittedly, we still only cover some of the possible information distributions since, for example, those with one or multiple plateaus are not represented. However, as these templates can no longer be adequately characterized by a scalar information slope  $\beta$  and these

distributions may only be possible for larger numbers of informational bits, this is outside the scope of this paper. Based on these templates, a proto-instance is generated by randomly sampling each integer at index  $i$  with the corresponding number of bits  $m_i$  from  $[2^{m_i-1}..2^{m_i})$  (cf. Van den Berg and Adriaans, 2021).

The fractal dimension of an SSP proto-instance is then measured on the histogram of all corresponding subset-sum frequencies, as shown in Figure 2. We use standard box-counting, where the slope of multiple box-counts in a log-log plot yields the Minkowski dimension (Bishop and Peres, 2016). Others methods, such as variational box-counting (Pilgrim and Taylor, 2018) or Haar wavelet approach (Zelinka et al., 2014), may not be feasible or at least not as easy to implement for large and sparse histograms resulting from larger proto-instances. A box is placed for every grid cell that the histogram covers, regardless by how little. The box-sizes are limited by the histogram to  $\{2^n | n \in \{0..\lceil \log_2 \max t \rceil\}\}$  bins. Box-counts are only performed for up to 10 different box-sizes to avoid excessive runtimes.

By applying the box-counting method to a binary normalization of the histograms, we obtain a dimension between 0 and 1 which is the Minkowski dimension in one dimensional space. We call this the line-counting dimension of a proto-instance, and it indicates the distribution of solvable values of  $t$  for a proto-instance. For a set of binary numbers or any other superincreasing set, the value for the box- and line-counting dimension are identical because each bin in the histogram has a value of at most 1.

Figure 2 also shows that a linear fit may not perfectly describe the scaling of the histogram. Thus, the *goodness of the fit* is computed using the coefficient of determination  $R^2$ . In our experiments, the fractal dimension is considered to be accurately characterized if  $R^2 \geq 0.95$ .

In experiment #1, we measure the fractal dimension using these methods for random proto-instances with different ratios of  $m/n$ , proportional to  $\alpha$  in Equation (1), with  $n = 15$  and uniform distribution of bits

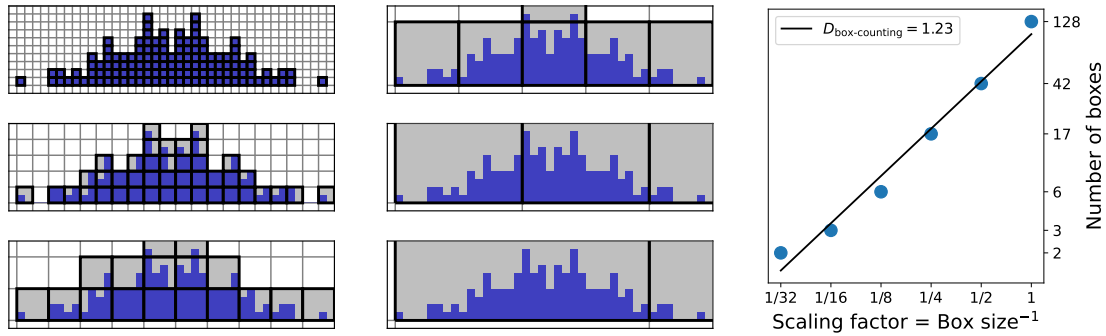


Figure 2: Box-counting method applied to the histogram of proto-instance of the SSP like those in Figure 1. This example uses a multiset  $S = \{4, 4, 5, 5, 6, 7, 7\}$  for better visualization. The right subfigure shows the log-log plot of the number of boxes touching the histogram in the other subfigures at different sizes. The resulting slope gives the fractal dimension.

( $T_{\beta \approx 0}$ ), inspired by Korf (1998) and Hayes (2002). In experiment #2, we measure the fractal dimension of proto-instances with the same number of bits over  $n = 15$  integers but with 39 different informational bit distribution slopes  $\beta$ , which were generated as described earlier. For each of these experiments, five proto-instances are created for every template, resulting in a total of 500 and 195 proto-instances respectively.

In addition to the fractal dimension of proto-instances, we also investigate their hardness by counting the number of recursions required by the depth-first BB algorithm by Van den Berg and Adriaans (2021). While other exact algorithms for the SSP might have a slightly better (yet still exponential) time complexity (Howgrave-Graham and Joux, 2010), this algorithm is exceptionally well suited for solving many instances in parallel due to its minimal spatial complexity. For any set  $S$ , there are obviously trivial instances, for example  $t \in S$ , and both easy and hard values of  $t$  are not the same between different sets of integers. Thus, we resort to measuring the hardness of the NPP where  $t = \lceil \frac{1}{2} \sum S \rceil$  for all sets like Van den Berg and Adriaans (2021) and consider  $\sum A = t + 1$  a perfect solution for all instances where  $\sum S \equiv 1 \pmod 2$ , like Schreiber et al. (2018).

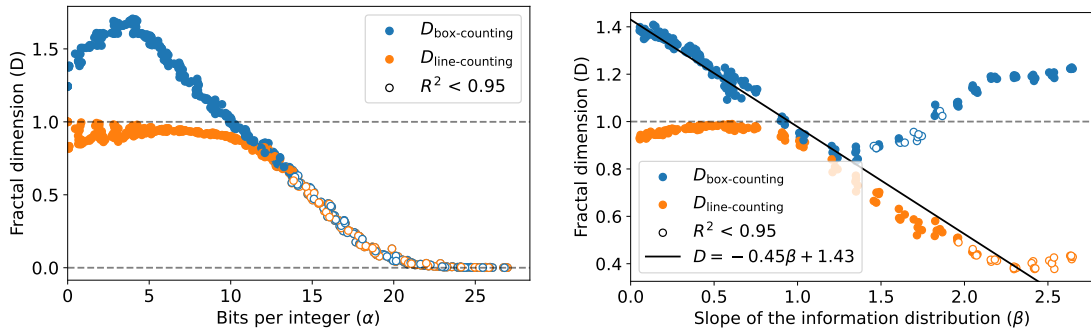
### 3 RESULTS

As expected, in both experiments, we obtain dimensionalities in the range of  $[0, 1]$  for the line-counting and  $[0, 2]$  for the box-counting method. For experiment #1 concerning different values for  $\alpha$  in Equation (1), almost exclusively those with  $\alpha \leq n$  (corresponding to  $m/n \leq 1$ ) have an  $R^2$  score of at least 0.95, which is also the range containing multiple solutions for instances of the NPP (Hayes, 2002). In total, 266 of the 500 sets have an  $R^2$  score of at least 0.95 for both box- and line-counting and are thus considered characterized by their

respective fractal dimension. The values for the line-counting dimension are within the range  $(0.644, 1]$ , with lower values of  $\alpha$  seeming to coincide with a higher maximum dimensionality. However, an incomplete arch with a peak at  $\alpha \approx 7$  is visible in Figure 3a.

Similarly, the box-counting dimension peaks at  $\alpha = 3.985$  with 1.706. Above and below this point, the values seem to decrease linearly. Generally, the fewer bits per integers, the less likely it is that there are *holes* for any values of  $t$  and so the line-counting dimension increases. At the same time, the box-counting dimension also increases with the frequency of subsets with the same sum. One could say that the histograms start having a not insignificant *height* across multiple values of  $t$ . Beyond a certain point, the number of bits is low enough that all possible values are likely represented. Say  $m/n \approx 0.26$ , then there are only  $2^{0.26n} \approx 2^4 = 16$  possible integer values. This may explain the lower box-counting dimension, since the small size of the histogram is once again covered by relatively few boxes of any size.

Figure 3b shows the box- and line-counting dimension of the generated SSP proto-instances over their corresponding information distribution, and looks somewhat similar to Figure 3a. Out of the 195 sets, 147 have an  $R^2$  scores of at least 0.95 for both box- and line-counting. The horizontal axis marks the slope of the information distribution for each instance, from flat at 0 and non-eccentric over linearly increasing at 1 to increasingly eccentric. The box-counting dimension increases for values left of the linearly increasing template with  $\beta = 1$  while the line-counting dimension decreases right of this value. We fit a line with a slope of  $-0.45$  and offset 1.43 through those points with  $R^2 \geq 0.98$ . Since  $m/n = \frac{8}{15} \approx 0.53$ , the line-counting dimension of the non-eccentric instances is very high and probably almost all values of  $t$  have a solution. Proto-instances with  $\beta > 1$  have some relatively large values, so the histogram of subset sum values is



(a) Box- and line-counting dimension over different bits per integer  $\alpha$ . For values above 15 ( $m/n \geq 1.0$ ), the object appears less fractal, as indicated by the  $R^2$  score.

(b) Values over different information distributions  $\beta$ . There are 120 bits over all integers.

Figure 3: Fractal dimension of the SSP proto-instances with  $n = 15$  integers over the  $\alpha$  in (a) and  $\beta$  in (b) of the information model Equation (1).

rather sparse, since its range is large. As this value is increased further, the clusters of possible subset-sum values  $t$  are stretched further apart, since the range increases. The box-counting method is not applicable for such instances because, in the extreme, the clusters of values look like two towers, two lines which stand orthogonally on the horizontal axis and so any fractal characteristic is not captured by a linear model over the entire range of box-sizes. With decreasing values of  $\beta$  the box-counting histogram becomes more and more bell shaped and similar to an Irwin-Hall distribution (Hall, 1927) at  $\beta = 0$ , because using a flat template with uniform distribution both very low and very high sums are rare. Thus, the box-counting dimension increases with the height of the histogram.

The change of the trend in the data for  $\beta \geq 2$  can perhaps be attributed to the fact that very eccentric proto-instances, due to the limitation on the number of bits, are inevitably trivial multisets containing mostly 1s. This may not occur so pronounced when increasing the number of bits representing the small numbers, however, changing the number of bits in an instance also affects the fractal dimension (cf. Figure 3a).

Figure 4 shows the hardness of the corresponding NPP instances, the number of recursions for the BB algorithm by Van den Berg and Adriaans (2021), for both experiments on a logarithmic scale. For sets from experiment #1 with a uniform distribution of bits ( $\beta = 0$ ) in the left subplot over  $\alpha$ , the number of recursions increases on average exponentially until around 1, mirroring the visualization of the number of optimal partitions (Hayes, 2002), recently characterized by Horn et al. (2024a). For the sets from experiment #2 with varying  $\beta$ , this is less clear-cut. Non-eccentric instances (low value of  $\beta$ ) are easier, but also more varied than eccentric ones, in line with

previous findings by Van den Berg and Adriaans (2021) and Sazhinov et al. (2023).

The Pearson correlation of the hardness of the generated NPP instances over  $\alpha$  with the box- and line-counting dimension is  $-0.666$  and  $-0.688$  respectively, while over  $\beta$  it is  $-0.751$  and  $-0.645$ . Although the correlation between the hardness and the box-counting dimension is higher over varying  $\beta$  than over varying  $\alpha$ , Figure 3a clearly shows that the latter develops much more similarly to the hardness for  $0.351 \leq m/n \leq 1$  ( $3.785 \leq \alpha \leq 13.810$ ) with a correlation coefficient of  $-0.834$  when using the logarithm of the number of recursions.

## 4 THE BROKEN ELEVATOR IN HILBERT'S HOTEL

Our investigations of the solution landscape concern instances of finite size and therefore finite complexity. *Dilation theory* (Adriaans, 2021) studies computable mappings of infinite sets onto higher dimensional discrete spaces. For this purpose, we consider  $\mathfrak{P}(\mathbb{N})$  the set of finite subsets of  $\mathbb{N}$ . It is *countable*, in contrast to the power set  $\mathcal{P}(\mathbb{N})$ , which is *uncountable*. Below, we illustrate how the complex structures that we observed before and the computational hardness of combinatorial problems are intricately linked.

The famous thought experiment of Hilbert's hotel, named after the German mathematician who introduced it in 1924 (Kragh, 2014), describes a hotel with an infinite number of rooms which is fully booked for a conference of mathematicians and yet can still accommodate other guests. We propose a new thought experiment starting from a hotel having infinitely many elevators (columns) which each lead

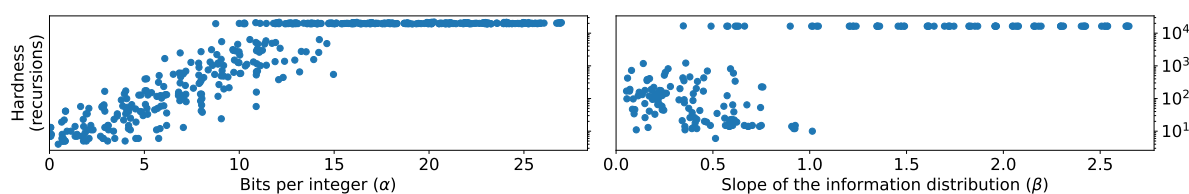


Figure 4: Hardness of the corresponding NPP instances for all generated sets over the number of bits per integer  $\alpha$  of a flat template and different information distributions  $\beta$  from Equation (1). The hardness of an instance is measured in recursions required by the BB algorithm by Van den Berg and Adriaans (2021).

to the  $i^{\text{th}}$  room on all infinite floors (rows), thus transforming the problem from  $\mathbb{N}$  into  $\mathbb{N}^2$ . Contrary to the original idea of Hilbert, we are interested in the variations of the *occupancy rate* of the hotel under various mathematical functions used to allocate the same countable number of guests. We imagine that all mathematicians who stay at the hotel are identified by a unique finite set of natural numbers such as  $\{1, 2, 5, 7\}$ ,  $\{n | n < 10!\}$  or  $\emptyset$ , effectively defining a bijection between the infinite set of mathematicians and the countable infinite set of finite subsets of natural numbers. When asked to allocate rooms for the participants of a conference, Hilbert decides to give each guest two functions to compute their room based on the set of numbers assigned to them:

- the *elevator index* function (column) and
- the *floor index* function (row).

Together these functions give the complete information of the location of the room in the hotel: the elevator index function gives a *partial description* of the location, the floor index function gives the *missing information*. Hilbert soon realizes there is a complex interaction between the elevator- and the floor index function. The occupancy rate of the hotel depends on the functions the guests use in order to compute the location of their rooms. When the guests use very little information from their set of numbers (e.g., the *cardinality* of the set) to select the elevator, the hotel is fully booked (defining a two-dimensional space). It is easy to compute the corresponding floor index. When the guests use all the information of the numbers in their sets (i.e., interpret them as integer selection masks for *binary numbers*) to select the elevator, the hotel is almost empty: only the first floor rooms are occupied (defining a one-dimensional space). For elevator index functions in between these two extremes, like *addition* and *multiplication*<sup>2</sup>, the behavior of the floor index function is chaotic and leads to a fragmented occupation of the hotel (see Figure 5). These chaotic regions

<sup>2</sup>Addition and multiplication are associated with the partition function estimated by Hardy and Ramanujan (1918) and integer factorization problem (Lenstra, 2011) respectively.

are associated with fundamental issues in mathematics, such as factorization and the partition function.

Functions in this region do not allocate guests very efficiently. At the conference, a young mathematician discovered this the hard way when she decided to organize a sub-conference and invite all  $2^n - 1$  colleagues she knows, that is to say, those whose set is a subset of hers. Since it is a small conference, she uses addition for the *elevator*-function so that all guests have rooms close to the lobby. She leaves the choice of the floor to each participant, depending on arrival. A day before the conference, she discovers that the elevator with the number 141592653 is out of order. She has to check if this elevator is used at all during the conference, but she soon realizes she has no efficient algorithm to answer this question, because it is an instance of the *NP*-complete SSP.

Likewise, with addition as the allocation function it is not easy for attendees to know if they will have direct neighbors on their floor to talk to, since knowing any given subset sum does not necessarily say anything about the frequency or existence of directly neighboring subset sum values. The take-away message of this example is that not all efficiently computable elevator index functions lead to efficiently computable floor index functions, which is the defining characteristic of the class *NP*. Dilation theory predicts that there is a correlation between the fractal dimension of the occupancy rate of Hilbert's Hotel associated with various elevator index functions and the hardness of the corresponding problems in *NP* that use these elevator index function as a *checking function*. The empirical research described in Section 2 and 3 corroborates that theory. The template approach presented in the first part of the paper was developed to study phase transitions in these chaotic regions of the SSP (see Figure 5). The previously measured fractal dimension can be seen as a measure of how the conference occupies the hotel. If the set is very eccentric, all subset sums do not occur more than once. All attendees are staying on the first floor, rather far apart from each other.

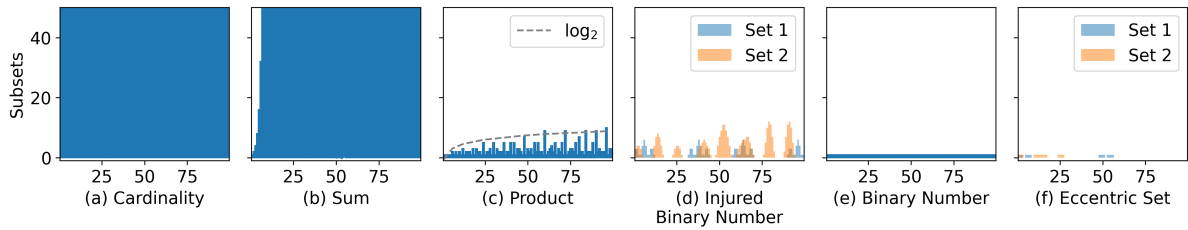


Figure 5: Different (truncated) histograms for functions defined on  $\mathfrak{P}(\mathbb{N})$ . The images (a), (b), (c) and (e) are associated with standard mathematical functions. The image (d) is a so-called injured set, and (f) shows an eccentric set. These two are designed to give insight in the behavior of sets in the neighborhood of the scale-free set defined by powers of two. The possible solution spaces of instances of the SSP are captured by (d)-(f).

## 5 DISCUSSION

Looking at Figure 5(b) and (d)-(f), we might relate them to different templates from the previous experiment. The set  $S = \{a..a + n - 1\} \subset \mathbb{N}$  for a relatively large number  $n = |S|$  is the *flattest* SSP proto-instance possible without including multisets, and there is an exponential approximation function for  $a = 1$  which describes the number of solutions for every possible value of  $t$  (Hardy and Ramanujan, 1918). As visualized in Figure 5 (b), its dimension is close to 2. Increasing the range of the integers but keeping the number of them constant results in a new instance corresponding to Figure 5 (d), (e) or (f) depending on the increase and distribution of the integers across the range of possible values as described before. The same number of possible subsets  $2^{|S|}$  is now stretched over a greater width, not unlike the stretching of a rubber band. As this *rubber band* flattens and finally tears, the fractal dimension decreases below 1 in Figure 5 (e), coming ever closer to 0.

The representation of a problem as counting the elements in the set  $\mathfrak{P}(\mathbb{N})$  is not limited to the trivial mathematical functions described here and visualized in Figure 5 and may be applied to the traveling salesman problem (TSP) (Hoffman and Padberg, 2001) with integer distances, the maximum satisfiability problem (MAX-SAT) (Bacchus et al., 2021) or instances of other discrete combinatorial optimization problems. For the TSP with integer distances, such a mapping would correspond to the set of selected numbered edges which make up the tour and the checking function being the total (integer) distance of the tour or zero, if the tour is invalid. For the MAX-SAT the mapping would correspond to the numbered variables assigned  $\top$  with the number of satisfied clauses as the checking function. In both cases, one can already see that there might be alternative representations, like expanding the satisfiability problem such that one integer represents multiple variables that are assigned  $\top$ . This will undoubtedly result in different

topological neighborhoods, like the diagonalization of the binary numbers, which *folds* the representation of  $\mathfrak{P}(\mathbb{N})$  in Figure 5 (e) to completely fill  $\mathbb{N}^2$ .

The SSP belongs to the class of NP-complete problems, because there is no known algorithm which can solve the hardest instance in polynomial time. In the worst case, we have to traverse most of the search tree just to find out that the subset with sum  $t$  does not exist. Due to the fractal property of the solution landscape, the concrete values  $t$  of the *no instances* for a given proto-instance  $S$  seem unpredictable. (Or at least the frequency of such subset sums seems unpredictable.) The issue of separability of yes- and no instances therefore seems like an important step towards understanding the true hardness of the SSP.

## 6 CONCLUSION

In Section 1 we revisited the SSP. In previous work (Van den Berg and Adriaans, 2021; Sazhinov et al., 2023), the statistical properties of its NPP variant have been investigated, but the solution landscape of the SSP was not explored. Its *fractality* may even have practical implications, e.g. to assess the feasibility of resource re-allocation (solution count for changed  $t$ ).

The countability of the set  $\mathfrak{P}(\mathbb{N})$  of finite subsets of  $\mathbb{N}$  has been demonstrated through the mapping onto different checking functions for basic mathematical operations in  $\mathbb{N}^2$  and visualized in Figure 5. The cardinality of the sets completely fills the plane, while the binary numbers result in a continuous one dimensional line. Between these two extremes we find operations with a non-integer dimension and which thus have a fractal landscape. For the SSP this is empirically investigated and supported by the findings in Section 2 and 3. The topology itself may not be the only source of complexity. The line of binary numbers can be re-mapped to  $\mathbb{N}^2$  by diagonalization, but the resulting topological neighborhoods are vastly different from those resulting from the

cardinality. If we have multiple equally valid options of mapping  $\mathfrak{P}(\mathbb{N})$  onto  $\mathbb{N}^2$ , perhaps this set should be called semi-countable. We have come up with a thought experiment in Section 4 that is an extension of Hilbert's Hotel in which mathematicians (represented by sets of numbers) need to be assigned to the rooms using elevator and floor numbers. Applying it to the addition operation of finite sets yields the subset sum histograms we analyzed using the fractal dimension.

Since the instances used in our experiments are quite small, it may be worth repeating them with larger ones. To achieve the same bit distribution, however, the total number of bits must increase accordingly, requiring custom data representations exceeding basic 64-bit primitives. Future work should also investigate more complex checking functions for the selection of columns, such as those discussed in Section 5. Representing instances of these problems as sets of natural numbers may not be a trivial task. We suspect that there are an infinite number of possible functions with no shared intrinsic information in their (possibly also fractal) structure. How does this reflect on the relation between problems in  $P$  and  $NP$ ? This task looks quite challenging, yet simultaneously promising.

## ACKNOWLEDGEMENTS

Computational resources (HPC-cluster HSUpper) have been provided by the project hpc.bw. hpc.bw is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU.

## REFERENCES

- Adriaans, P. (2021). Differential information theory.
- Bacchus, F., Järvisalo, M., and Martins, R. (2021). *Maximum Satisfiability*, chapter 24. IOS Press.
- Bishop, C. J. and Peres, Y. (2016). *Minkowski and Hausdorff dimensions*, page 1–44. Cambridge Studies in Advanced Mathematics. Cambridge University Press.
- van den Berg, D. and Adriaans, P. (2021). Subset Sum and the Distribution of Information. In *Proceedings of the 13th International Joint Conference on Computational Intelligence*, pages 134–140.
- Ewald, W. and Sieg, W. (2013). Lectures on the infinite. In *David Hilbert's Lectures on the Foundations of Arithmetic and Logic 1917-1933*, pages 655–785. Springer Berlin Heidelberg.
- Falconer, K. (2013). *Fractals: A Very Short Introduction*. Oxford University Press.
- Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA.
- Hall, P. (1927). The distribution of means for samples of size  $N$  drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika*, 19(3-4):240–244.
- Hardy, G. H. and Ramanujan, S. (1918). Asymptotic formulae in combinatory analysis. *Proceedings of the London Mathematical Society*, s2-17(1):75–115.
- Hayes, B. (2002). Computing Science: The Easiest Hard Problem. *American Scientist*, 90(2):113–117.
- Hoffman, K. L. and Padberg, M. (2001). *Traveling salesman problem*, page 849–853. Springer US.
- Horn, R., Thomson, S. L., van den Berg, D., and Adriaans, P. (2024a). The easiest hard problem: Now even easier. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '24 Companion*, page 97–98, New York, NY, USA. Association for Computing Machinery.
- Horn, R., van den Berg, D., and Adriaans, P. (2024b). Fractal analysis of the subset-sum problem. <https://anonymous.4open.science/r/fractal-ssp>. (Replication package).
- Howgrave-Graham, N. and Joux, A. (2010). *New Generic Algorithms for Hard Knapsacks*, page 235–256. Springer Berlin Heidelberg.
- Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA.
- Kleinberg, J. and Tardos, E. (2005). *Algorithm Design*. Pearson, Upper Saddle River, NJ.
- Korf, R. (1998). A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106(2):181–203.
- Kragh, H. (2014). The True (?) Story of Hilbert's Infinite Hotel.
- Lenstra, A. K. (2011). *Integer Factoring*, page 611–618. Springer US.
- Merkle, R. and Hellman, M. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530.
- Mertens, S. (2003). The easiest hard problem: Number partitioning.
- Pilgrim, I. and Taylor, R. P. (2018). Fractal Analysis of Time-Series Data Sets: Methods and Challenges. In Ouafeul, S.-A., editor, *Fractal Analysis*, chapter 2. IntechOpen, Rijeka.
- Sazhinov, N., Horn, R., Adriaans, P., and van den Berg, D. (2023). The partition problem, and how the distribution of input bits affects the solving process. In *Proceedings of the 15th International Conference on Evolutionary Computation Theory and Applications*.
- Schreiber, E. L., Korf, R. E., and Moffitt, M. D. (2018). Optimal Multi-Way Number Partitioning. *J. ACM*, 65(4).
- Sharma, S., Sharma, P., and Dhakar, R. S. (2011). RSA algorithm using modified subset sum cryptosystem. In *2011 2nd International Conference on Computer and Communication Technology (ICCCCT-2011)*, pages 457–461.
- Zelinka, I., Zmeskal, O., and Saloun, P. (2014). *Fractal Analysis of Fitness Landscapes*, pages 427–456. Springer Berlin Heidelberg, Berlin, Heidelberg.