Notre Dame Journal of Formal Logic
Volume 57, Number 1, 2016

Controlling Effective Packing Dimension of Δ^0_2 Degrees

Jonathan Stephenson

Abstract This paper presents a refinement of a result by Conidis, who proved that there is a real X of effective packing dimension $0 < \alpha < 1$ which cannot compute any real of effective packing dimension 1. The original construction was carried out below $\emptyset''$, and this paper's result is an improvement in the effectiveness of the argument, constructing such an X by a limit-computable approximation to get $X \leq_T \emptyset'$.

1 Introduction

Effective packing dimension is one of several common ways to study the way in which information is encoded in a real number. It assigns to each $X \in 2^\omega$ a real number $\dim_P(X) \in [0, 1]$. The concept has been considered in a wider context in a number of other publications (see, e.g., Downey and Hirschfeldt's book [5], Downey and Greenberg [4], and Downey and Ng [6]). An effective packing dimension equal to 1 corresponds to the notion that infinitely many of the initial segments of the real are unable to be significantly compressed by any algorithmic process, whereas an effective packing dimension of zero indicates that initial segments of the real number are easily deduced from a relatively small amount of information. Martin-Löf random reals have effective packing dimension 1, and in fact are characterized as the class of reals for which all initial segments are largely incompressible; they are a well-studied class of reals, and have several other natural characterizations. At the other end of the spectrum of effective packing dimension are reals with effective packing dimension 0. Included in this class are both computable reals and noncomputable K-trivial reals which encode information, but in a very sparse manner.

The strong links between algorithmic randomness, information content, and effective packing dimension lead to reals whose effective packing dimensions lie strictly

Received February 23, 2013; accepted September 29, 2013
First published online November 16, 2015
2010 Mathematics Subject Classification: Primary 03D32; Secondary 68Q30
Keywords: effective packing dimension, limit-computable approximation, complexity, pruned clumpy trees
 10.1215/00294527-3328401

between 0 and 1 being often regarded as being "partially random." This notion of partial randomness does not respect the Turing degree structure particularly well, because one can easily exhibit reals in every Turing degree with effective packing dimension 0. Indeed, given a real Z of effective packing dimension $\alpha > 0$ and some rational number q with $0 \leq q < \alpha$, it is easy to produce a real in the same Turing degree as Z with effective packing dimension less than q, by "thinning" the information coded by Z in a computable way.

The goal of this paper is to improve on a result of Conidis [3], who constructed a real X which has intrinsically intermediate effective packing dimension in the following sense.

Theorem 1.1 (Conidis [3, Theorem 6.1]) *There is a real $X \leq_T \emptyset''$ with nonzero effective packing dimension such that X cannot compute any reals of effective packing dimension* 1.

The main theorem of this paper increases the effectiveness of this result by constructing a real with the same property, but which is below $\emptyset'$ in the Turing degree structure.

Theorem 1.2 *There is a real $X \leq_T \emptyset'$ with nonzero effective packing dimension such that X cannot compute any reals of effective packing dimension* 1.

The reals constructed by this theorem and that of Conidis are not able to be obtained by applying the "thinning" process described above to a real with effective packing dimension 1 and which is in the same Turing degree.

We will obtain a corollary by applying one of the theorems of Bienvenu, Doty, and Stephan [2, Theorem 2.5], who repackage a result of Fortnow, Hitchcock, Pavan, Vinodchandran, and Wang [7, Theorem 5.2], stating that any Turing degree which contains a real of nonzero effective packing dimension contains reals of effective packing dimension arbitrarily close to 1. Recall that the effective packing dimension of a set of reals is equal to the supremum of the effective packing dimensions of its members, and consider the Turing degree of the real X of the previous theorem, to deduce the following.

Corollary 1.3 *There is a Δ^0_2 Turing degree with effective packing dimension* 1 *which contains no real of effective packing dimension* 1.

It is worth noting that effective packing dimension is only one of the ways to measure the complexity of a real number. A better-known measure is the effective Hausdorff dimension, introduced by Lutz [10], and which has been well studied; in particular, some questions analogous to those studied here are considered in papers by Greenberg and Miller [8], and Miller [11], and (once again) a good overview of the area is given by Downey and Hirschfeldt [5].

Both the means of construction and the improvements in effectiveness seen in this paper and that of Conidis [3] correspond very closely to that which occurred in the development of minimal Turing degrees in classical computability theory. In that case the original construction of a minimal degree was carried out by Spector in 1956 (see [15]), with the construction occurring below $\emptyset''$. In 1961, Sacks [12] noted that the construction could also be carried out below $\emptyset'$. In effect, each of the $\emptyset''$-oracle questions was replaced by sequences of $\emptyset'$-computable questions.

This bound was later improved by Yates [16] in a 1970 paper which showed that a minimal degree could be found below any c.e. degree. In this case the method used was a limit computable (full approximation) construction.

In each case the construction is given by finding a sequence of computable trees T_i with the property that for each i, $T_{i+1} \subseteq T_i$, and so that for each T_i and each path X through T_i such that Φ_i^X is a total reduction, either Φ_i^X is computable, or it is a real which computes X. Thus the unique real X which is a path through each of the T_i is in a degree which is minimal with respect to Turing reductions.

The construction given for reals with nonzero effective packing dimension and which cannot compute any reals of effective packing dimension 1 in Conidis [3] and that presented here follow a similar developmental history. As in the case of the minimal degree arguments, the constructions are carried out by building a nested sequence of trees, chosen so that the unique path which lies in all of the trees satisfies the requirements of the problem.

The original proof by Conidis, which constructed a degree below $\emptyset''$, is herein replaced by an approximation by $\emptyset'$-computable procedures, yielding a Δ_2^0 real with the desired properties. It should be noted that the further improvement given by Yates in the case of minimal degrees which allowed that construction to be carried out below any c.e. degree will not have an analogy in our case, since there are noncomputable degrees which cannot compute any real with nonzero effective packing dimension (e.g., the K-trivial degrees). The question of exactly which Turing degrees can compute a real of nonzero effective packing dimension that cannot in turn compute a real of effective packing dimension 1 remains an interesting and relevant one, for which some thoughts and a conjectured partial solution are given in Section 7.

Some of the technical machinery and lemmas used by Conidis [3] will be useful here. The notation of that paper is followed wherever it is reasonable to do so. For $\sigma, \tau \in 2^{<\omega}$ write $\sigma\tau$ to indicate the string formed by concatenating σ and τ. Let $A, B \subseteq 2^{<\omega}$. Write AB to mean $\{\sigma\tau : \sigma \in A, \tau \in B\}$, and similarly $A\sigma$ and σA for $A\{\sigma\}$ and $\{\sigma\}A$, respectively. I denote by $2^{\leq n}$ the set of strings in $2^{<\omega}$ of length at most n, and by $2^{=n}$ those of length exactly equal to n. Write $|\sigma|$ for the length of a string $\sigma \in 2^{<\omega}$.

By $\sigma \preceq \tau$ I mean that σ is an initial segment of τ, and by $\sigma \prec \tau$ that $\sigma \preceq \tau$ and $\sigma \neq \tau$.

I will use K to denote prefix-free Kolmogorov complexity of strings. This will be the only notion of complexity considered throughout the paper; for brevity I simply refer to K as Kolmogorov complexity. It should be noted that the usage of prefix-free complexity here is not important to the construction: plain Kolmogorov complexity could be used instead, yielding the same results.

As pointed out by Conidis [3], the following definition of effective packing dimension is not the standard one, but the proof that they are equivalent is indicated by Athreya, Hitchcock, Lutz, and Mayordomo [1], noting that it follows from a similar result by Lutz [10].

Definition 1.4 Let $X \in 2^{\omega}$. The *effective packing dimension of* X is

$$\dim_P(X) = \limsup_{n \to \infty} \frac{K(X \upharpoonright n)}{n}.$$

If $\mathbf{d} \subseteq 2^\omega$ is a Turing degree, then the *effective packing dimension of* $\mathbf{d}$ is

$$\dim_P(\mathbf{d}) = \sup_{X \in \mathbf{d}} \limsup_{n\to\infty} \frac{K(X \upharpoonright n)}{n}.$$

Definition 1.5 Let $\{\Phi_i\}_{i\in\omega}$ be a computable listing of all oracle Turing machines. By an oracle Turing machine Φ_e we will mean one which outputs 0 or 1 as a result of any halting computation.

Notation shall otherwise be as in Soare [14] and [13]. In particular, by $\Phi^\rho_{e,s}(x) = y$ I mean that the oracle computation $\Phi^\rho_e(x)$ halts with output y within s steps of computation, and with use no greater than $|\rho|$. I assume that computations of this kind either halt at some stage $s < |\rho|$, or not at all, and that they may halt only if $x < |\rho|$, so that finite oracle strings only compute finitely many outputs.

Adopting the conventions used by Conidis, assume that the oracle computations are monotonic in their use, in the sense that if $\Phi^\rho_{e,s}(x)\downarrow$ and $y < x$, then $\Phi^\rho_{e,s}(y)\downarrow$ too. This convention will ensure that if $\rho \in 2^{<\omega}$, then there is a string $\tau \in 2^{<\omega}$ with $\Phi^\rho(x)\downarrow$ precisely when $x < |\tau|$ and that $\Phi^\rho_i(x) = \tau(x)$ in this situation. We will denote this by $\Phi^\rho_i = \tau$ (notice that τ is the longest string computed by ρ in this way). We will also assume a second monotonicity condition that if ρ is obtained from a string τ by extending it by a single 0 or 1, then there is at most one x such that $\Phi^\rho_i(x)\downarrow$ but $\Phi^\tau_i(x)\uparrow$. This will ensure that if T is a tree, then defining $\Phi^T_i = \{\tau \in 2^{<\omega}: (\exists\rho \in T)\Phi^\rho_i = \tau\}$, Φ^T_i will also be a tree; it is worth noting that given a c.e. tree T and an enumeration for T, we can effectively enumerate Φ^T_i.

Definition 1.6 A *clump* is a subset of $2^{<\omega}$ of form $\sigma 2^{\leq|\sigma|}$, where $\sigma \in 2^{<\omega}$. We refer to σ as the *root* of the clump.

Definition 1.7 A *pruned clump* is a subset of $2^{<\omega}$ which is a downward closed subset of a clump $\sigma 2^{\leq|\sigma|}$, and which contains at least two of the leaves of $\sigma 2^{\leq|\sigma|}$. Once again, refer to σ as the *root* of the pruned clump.

Pruned clumps will be the main tool used to make sure that the real we will construct has nonzero packing dimension, and will consequently play a major role in the construction which forms Section 4 of this paper.

Definition 1.8 An *extendible string* on a tree T is an element $\tau \in T$ with the property that there is a path through T which has τ as an initial segment.

Definition 1.9 A *pruned clumpy tree* is a tree $T \subseteq 2^{<\omega}$ such that whenever $\tau \in T$ is an extendible string on T, there is a string $\sigma \in T$ such that $\tau \prec \sigma$ and a pruned clump with root σ is on T.

2 Strategy

The construction which is laid out in this paper will build a sequence of c.e. pruned clumpy trees $T_0 \supset T_1 \supset T_2 \cdots$, choosing a string $\xi_i \in T_i$ for each i. We will construct the ξ_i so that for each i, j either $\xi_i \prec \xi_j$, or vice versa. Therefore the ξ_i will specify a unique path X which lies in the intersection of the T_i. The ξ_i will also be chosen so that for each i, ξ_i has some initial segment ρ of length at least i, and which has high Kolmogorov complexity, which will ensure that the path X will have nonzero effective packing dimension (indeed the dimension will be at least $\frac{1}{4}$). The trees T_i are chosen to have enough leaves on each clump that such strings ρ exist, but the leaves which are added to the clumps are chosen carefully, so that X does not compute any real of effective packing dimension 1. We will see that this is done by either forcing that Φ_i^Y is nontotal for each path Y through T_i, or that there is some $\alpha < 1$ which is an upper bound on the effective packing dimension of the real Φ_i^X. This latter case is achieved by a process of *majority vote*: by pruning T_i carefully, we can ensure that a large number of the reals computed with paths through T_i as oracles agree on long initial segments, and thus guarantee that Φ_i^X is a path through a c.e. tree which branches sufficiently slowly that the initial segments of Φ_i^X must have quite low Kolmogorov complexity—we will see that identifying such an initial segment may be achieved by giving its length, together with information about its position in the lexicographical ordering of strings of that length on $\Phi_i^{T_i}$, which will require a number of bits proportional to the length of the segment, with a constant of proportionality less than 1.

The construction will proceed by constructing for each i a sequence $T_i^0, T_i^1, T_i^2, \ldots$ of finite trees, with the property that for each i and any $k < k'$, we have $T_i^k \subset T_i^{k'}$. We will ensure that

$$T_i = \bigcup_k T_i^k$$

is a computably enumerable pruned clumpy tree. In addition, we will $\emptyset'$-computably construct sequences ξ_i^k of strings which will converge to limits ξ_i. The string ξ_i will act as a root to the tree T_i in the sense that all extendible strings on T_i will either extend ξ_i or be initial segments of it. To ensure that there is a real $X \in 2^\omega$ which has each ξ_i as an initial segment, we will only ever let ξ_i^k be an extension of ξ_i^{k-1}, and ensure that every other ξ_j^k we construct is $\preceq$-comparable with ξ_i^k. Thus the ξ_i will be the limits of uniformly $\emptyset'$-limit-computable sequences. These strings will define a real X which is $\emptyset'$-computable.

In the construction given by Conidis [3, Section 5], the $\emptyset''$-oracle was able to identify whether a given tree T_{i-1} contained a string τ with the property that for some $x \in \omega$ and every extension $\rho \in T_{i-1}$ of τ, one had $\Phi_i^\rho(x) \uparrow$. This allowed detection of strings forcing each oracle computation Φ_i^Y with τ as an initial segment of a path Y in T_{i-1} to be nontotal. This will not be possible to achieve using only a $\emptyset'$-oracle.

At each stage k of the construction carried out in this paper we will construct a finite tree T_i^k for each $i \leq k$. We will also build a string σ^k with the property that $\sigma^k(i)$ tells us whether we currently believe that a string τ fulfilling a similar role to that in Conidis's construction exists on T_{i-1}^k. We will be hampered by two factors in this matter. First, we do not know what the final tree T_{i-1} will be while we are building T_i. For this reason we will work with the assumption that it will not contain such a string τ, until we are able to identify one. Thus we will build trees $\widehat{T}_i^{k,s}$ for each $k \geq s$, which will correspond to our current guess at T_i^s under this assumption. When k is large enough, we will have $\widehat{T}_i^{k,s} = T_i^s$ for all s. Second, we will not be able to consult $\emptyset''$ to ask whether any string on T_{i-1} forces divergence of an oracle computation. Identifying such a string will be achieved by approximating this question: we will ask only if a specific string forces divergence on some specific computation.

At each stage k of the construction, we will follow one of two strategies: either construct extensions which output halting computations under the ith oracle machine, or identify a string on our tree whose extensions all diverge under this oracle computation.

Remark 2.1 Let T be a computable tree, let $\tau \in T$, and consider the statement

$$(\exists x < n)(\forall \rho \in T)(\forall s \in \omega)\big[\tau \preceq \rho \to \Phi_{i,s}^{\tau}(x) \uparrow\big],$$

which asserts that τ forces divergence of the oracle machine Φ_i on all of its extensions on some (bounded) input x. This statement is uniformly computable in $\emptyset'$ given T, τ, n.

The portion of the strategy which seeks to force our real X to have nonzero effective packing dimension will make heavy use of the following lemma; it is put to the same use by Conidis [3].

Lemma 2.2 (Conidis [3, Lemma 3.2]) *Let $q \in \mathbb{Q}$, $0 \leq q \leq 1$, and $\tau \in 2^{<\omega}$, and let q_τ be the least natural number that is greater than or equal to $q|\tau|$. Then, for any given pruned clump of the form $A \subseteq \tau 2^{\leq|\tau|}$ such that A contains at least 2^{q_τ} many leaves of $\tau 2^{\leq|\tau|}$, there is a leaf $\sigma \in \tau 2^{\leq|\tau|}$ in A such that*

$$K(\sigma) > \frac{q}{2}|\sigma| - 1.$$

If X is the real we construct, and we consider some i such that Φ_i^X gives a total reduction, then the proof that $\dim_P(\Phi_i^X) < 1$ will rely on estimates on the initial segment prefix-free complexity of the real Φ_i^X obtained by considering a pair of prefix-free machines.

Definition 2.3 For the purposes of computing Kolmogorov complexity, a prefix-free machine M is a partial computable map from $2^{<\omega}$ to $2^{<\omega}$ with the property that if $\sigma \prec \tau$, then $M(\sigma) \downarrow$ implies $M(\tau) \uparrow$.

We will use the fact that if M is a prefix-free machine, then there is some constant C (depending on M) such that for each pair σ, τ of binary strings with $M(\tau) = \sigma$, we have $K(\sigma) < |\tau| + C$.

The idea will be to construct two machines M_1 and M_2 which, between them, exhibit short descriptions of the initial segments of Φ_i^X. They will do so by making efficient use of the structure we build into the trees we construct to extract information about strings computed from those trees.

The remainder of the paper is organized as follows. In Section 3 we will see two procedures which will be used to carry out the construction of the trees T_i which we seek. This construction is laid out in detail in Section 4. In Section 5 we will check that the construction yields c.e. trees and will see that there is a single path X common to all of the trees, with effective packing dimension at least $\frac{1}{4}$. This will be the path we seek. In Section 6 we will verify that the construction carried out will guarantee that any real computed by X must have effective packing dimension strictly less than 1, by giving an explicit bound via a combinatorial estimate.

3 Approximations

This section outlines the algorithms which will be used to compute finite approximations T_i^k to a tree T_i based on the trees T_j^k for $j < i$ at some stage $k \geq i$.

Suppose that we have a computable map from ω onto the set of all finite trees, that is, an indexing of those trees.

Now define two computable maps $\mathcal{T}: 2^{<\omega} \times \omega^2 \to \omega$ and $\mathcal{S}: \omega^4 \to \omega$, where we think of each map as outputting a finite tree. The maps $\mathcal{T}$ and $\mathcal{S}$ each provide a means to guess at what some of the trees T_i we are building in the overall construction are.

The algorithm $\mathcal{S}$ The algorithm $\mathcal{S}$ will be used at stages of the construction at which we are searching for strings to add to the finite approximations to our trees T_i in order to obtain longer halting computations. The computations which we seek will be required to come in families which all agree on some initial segments of fixed length; this will help satisfy the requirement that $\dim_P(\Phi_i^X) < 1$, where X is the real which we are constructing.

Let $\mathcal{S}: \omega^4 \times 2^{<\omega} \to \omega$ be the following algorithm. Let $\mathcal{S}(m, n, e, t, \xi) = p$, where p is an index for the finite tree R constructed as follows. Search for the leaves of the tree T with index m and which are extensions of ξ; note that this search terminates, since T is finite. Then for each leaf $\lambda_0 \succeq \xi$ of T, suppose that λ_0 is of length l. Attempt (by searching within the tree S with index n) to extend λ_0 to a string λ which is the root of a pruned clump on the tree S, of length $4^N > 4l \cdot 2^{2e+4}$ for some $N \geq e + 2$. For each λ_0, choose N to be the minimal possible, choosing the lex-least λ if a tiebreaker is needed. For each string λ obtained in this way, let L_λ be the leaves of the pruned clump on S with root λ. Now for each λ and each $\rho \in L_\lambda$, we search for a string $\widehat{\rho} \succeq \rho$ on S such that for each $x < 2^{-2e-4}|\lambda|$, we have $\Phi_e^{\hat{\rho}}(x) \downarrow$. Accept only strings $\widehat{\rho}$ for which $|\widehat{\rho}| < t$ and $\Phi_{e,t}^{\hat{\rho}}(x) \downarrow$ for each relevant x. Call such extensions *suitable with bound t*.

If there is no λ for which we find suitable extensions $\widehat{\rho}$ for each $\rho \in L_\lambda$, let $R = T$; we are not making any changes to the input tree. In this case, say that $\mathcal{S}$ *fails*; otherwise it *succeeds*.

If $\mathcal{S}$ succeeds, then for each λ such that we found suitable extensions for every $\rho \in L_\lambda$, find the least $i \leq t$ such that each $\rho \in L_\lambda$ admits a suitable extension with bound i, and for each such ρ let $\widehat{\rho}$ be the least suitable extension with bound i, when strings are ordered by length, and strings of the same length are ordered lexicographically. Let the set of all such $\widehat{\rho} \succeq \lambda$ obtained in this way be D_λ. Now choose the string τ_λ of length x for which we have

$$\left\{\widehat{\rho} \in D_\lambda : (\forall y < x)\left[\tau_\lambda(y) = \Phi_e^{\hat{\rho}}(y)\right]\right\}$$

of maximal size, choosing the lex-least such τ_λ in the case of a tie. Then add to R the downward closure of the nodes $\widehat{\rho} \in D_\lambda$ for which we have $\tau_\lambda(y) = \Phi_e^{\hat{\rho}}(y)$ for each $y < x$.

The algorithm $\mathcal{T}$ The algorithm $\mathcal{T}$ will be used at stages of the construction of T_i at which we have already found a string which we believe is not an initial segment of any path Y through T_{i-1} for which Φ_i^Y is a total map. In this case, we are simply keeping all extensions of that string.

Let $\mathcal{T}(\xi, m, n) = p$, where m, n are indices for trees T, S, respectively; let p be an index for the computable tree R given by letting R consist of precisely those strings τ such that either $\tau \in T$, or $\tau \in S$ and τ extends both ξ and some leaf of T.

4 Construction

In this section I construct a nested sequence $\{T_i\}_{i\in\omega}$ of computable trees and strings $\xi_i \in T_i$ which define a path $X \in 2^\omega$. This path will be computable in $\emptyset'$ and will not compute any reals of packing dimension 1, but will have nonzero packing dimension. The entire construction will be carried out below $\emptyset'$.

We will assume a default strategy for each i that in building the tree T_i, we will always find halting computations when we want them; we will move to our secondary strategy of forcing divergence if this ceases to be a viable strategy.

At stage k of the construction, we will choose a string $\sigma^k \in \{-1,0,1\}^{<\omega}$ of length $k+1$. For each $0 \le i \le k+1$, we will build a tree $T_i^k = T^k$ which will be our finite approximation to T_i, and a string ξ_i^k which will be our approximation to ξ_i. The string σ^k will tell us what strategies we are following at stage k of the construction the construction. If $\sigma^k(i) = -1$, we have not yet met the requirement that ξ_i^k has an initial segment of length at least i, and which is of high complexity. If $\sigma^k(i) = 0$, then we currently do not believe that we will be able to choose ξ_i in a way that the paths Y on T_i which pass through ξ_i all have Φ_i^Y nontotal. Otherwise $\sigma^k(i) = 1$, in which case we believe we have selected ξ_i^k to force nontotality of such computations.

We will ensure that ξ_i^k only changes at finitely many stages k, and therefore that it comes to a limit ξ_i. We will see that the ξ_i are all distinct strings which form a total order under inclusion as initial segments, and therefore that they define a unique $X \in 2^\omega$, which is the unique path through the T_i.

For each i, we will define

$$T_i = \bigcup_k T_i^k.$$

The resulting tree will be c.e. and have the property that for every k, each leaf of T_i^k extends some leaf of T_i^{k-1}.

For the base case, begin by setting $T_{-1}^0 = \{\emptyset, 0\}$, that is, a tree with a single leaf of length 1, and recursively letting T_{-1}^k be the downward closure of

$$\bigcup_{\substack{\lambda\in T_{-1}^{k-1}\\ \lambda \text{ a leaf}}} \lambda 2^{=|\lambda|} 0^{2|\lambda|}$$

so that each T_{-1}^k has leaves of length 4^k; this tree establishes the clumpy structure which underlies all of the T_i which we will construct. The specific choice of the root lengths for the pruned clumps here will ensure that some later computations will give integer bounds in cases where it is convenient that this should be so.

Now for the construction proper. At stage $k \ge 0$, for each $i < k$ we are given finite trees T_i^{k-1} and strings $\xi_i^{k-1} \in 2^{<\omega}$ as well as a string σ^{k-1} of length k. The trees given satisfy $T_{i-1}^{k-1} \subseteq T_i^{k-1}$ and $\xi_{i-1}^{k-1} \preceq \xi_i^{k-1}$ or $\xi_{i-1}^{k-1} \succeq \xi_i^{k-1}$ for $0 \le i < k$.

Proceed by a series of substages, one for each $0 \le i \le k$. At substage i we are given T_j^k, ξ_j^k, and $\sigma^k(j)$ for each $j < i$. We will build auxiliary computable trees $\widehat{T}_i^{k,s}$ for each $i < k-1$ as uniformly computable sets of strings. These trees will tell us which strings we currently believe are to be added to T_i at some stage. We will see that for each i and for large enough k, we have $T_i^s = \widehat{T}_i^{k,s}$ for each $s \ge k$, so that $\widehat{T}_i^{k,s}$ is best thought of as a prediction of what T_i^s will be, according to the information which is available at stage k. That the T_i^s are eventually equal to the $\widehat{T}_i^{k,s}$ will guarantee that the tree T_i which is constructed will be a c.e. set of strings.

Constructing T_{-1}^k This step in the computation serves to initialize the construction for stage k.

At each stage k, let T_{-1}^k be as described above, and let ξ_{-1}^k be the empty string. Define $\sigma^k(-1) = 1$, for convenience.

In addition, for each $s \geq k$, let $\widehat{T}_{-1}^{k,s} = T_{-1}^s$.

Constructing T_i^k for $0 \leq i < k$ In this section, suppose that we have built T_j^k and defined ξ_j^k and $\sigma^k(j)$ for each $j < i$, as well as having defined T_j^{k-1}, ξ_j^{k-1}, and $\sigma^{k-1}(j)$ for each $j < k$. We also assume that for $j < i-1$, we have built $\widehat{T}_j^{k,s}$ for each $s \geq k$.

Before building T_i^k, we will construct trees $\widehat{T}_{i-1}^{k,s}$ which give a uniformly c.e. guess at T_{i-1}. In the case that $i = 0$, we have already built the $\widehat{T}_{i-1}^{k,s}$, and can omit this step.

So suppose that $i > 0$, and that we are given $\widehat{T}_{i-2}^{k,s}$ for each s. Define $\widehat{T}_{i-1}^{k,k} = T_{i-1}^k$. Given $T_{i-1}^{k,s-1}$, let $\widehat{T}_{i-1}^{k,s-1}$ have index m, and let $\widehat{T}_{i-2}^{k,s}$ have index n, and set $\widehat{T}_{i-1}^{k,s}$ to be the tree given by $\mathcal{S}(m, n, i, s, \xi_i^k)$ if $\sigma^k(i-1) = 0$, and set $\widehat{T}_{i-1}^{k,s}$ to be the tree given by $\mathcal{T}(m, n, \xi_i^k)$ if $\sigma^k(i-1) = 1$.

First check whether any $j < i$ has $\sigma^k(j) = 1$ but $\sigma^{k-1}(j) = 0$, or any such j has $\sigma^k(j) = -1$. In this case our strategy for T_i has been interrupted, or is not yet active. So we must start over; to do so, set $\xi_i^k = \xi_i^{k-1}$, let $T_i^k = T_i^{k-1}$, and set $\sigma^k(i) = -1$. Then proceed to the construction of T_{i+1}^k.

Otherwise, if $\sigma^{k-1}(i) = 1$, then we know that ξ_i^{k-1} forces divergence as an oracle for the machine Φ_i on a tree $\widehat{T}_{i-1}^{k-1}$ whose construction was given earlier. In this case, let $\xi_i^k = \xi_i^{k-1}$, let $\sigma^k(i) = 1$, and let T_i^k be the tree given by $\mathcal{T}(m, n, \xi_i^k)$, where m and n are indices for the trees T_i^{k-1} and T_{i-1}^k, respectively. Then proceed to the construction of T_{i+1}^k.

If we have $\sigma^{k-1}(i) = -1$, we have not confirmed that $K(\rho) > \frac{1}{4}|\rho| - 1$ for some $\rho \preceq \xi_i^k$ with $|\rho| \geq i$.

If i is the least number with $\sigma^k(i) = -1$, use $\emptyset'$ to check whether there is a string ρ on T_{i-1}^k which is comparable to each ξ_j^k for each j such that $j < i$ and to ξ_j^{k-1} for each j such that $i < j < k$, and for which we have $K(\rho) > \frac{1}{4}|\rho| - 1$ and $|\rho| \geq i$. If such a string ρ exists, set ξ_i^k to be a leaf on T_{i-1}^k which extends ρ and set T_i^k to be equal to T_{i-1}^k, and set $\sigma^k(i) = 0$. If no such string exists, set $T_i^k = T_i^{k-1}$ and $\xi_i^k = \xi_i^{k-1}$, and $\sigma^k(i) = -1$. Then move to the construction of T_{i+1}^k.

Otherwise $\sigma^k(i) = 0$, and we have met the complexity condition for index i, and will use the approximation $\widehat{T}_{i-1}^{k,s}$ to T_j for each $j < i$ to check whether we can force divergence by moving to a new string ξ_i^k. Take m to be an index for T_i^{k-1}, n to be an index for T_{i-1}^k, and n^s to be an index for $\widehat{T}_{i-1}^{k,s}$, for each $s > k$. Check whether for some s, $\mathcal{S}(m, n^s, i, s, \tau)$ succeeds, where τ is the longest string in the set $\{\xi_{-1}^k, \xi_0^k, \xi_1^k, \ldots, \xi_{i-1}^k, \xi_i^{k-1}, \ldots, \xi_{k-1}^{k-1}\}$.

If it does succeed, let $\xi_i^k = \xi_i^{k-1}$ and T_i^k be $\mathcal{S}(m, n, i, k, \xi_i^k)$, and set $\sigma^k(i) = 0$, then proceed to the construction of T_{i+1}^k. Note that we are applying the algorithm $\mathcal{S}$ to a different string than the one which we checked for success on; this is because we cannot in general computably find the string τ of the previous paragraph, and will want to construct our tree by a method which is able to be computably approximated once our strategy has settled.

Otherwise, check whether there is some leaf λ_0 on T_i^{k-1} which extends every ξ_j^k for $j < i$ and every ξ_j^{k-1} for $i < j < k$, and which has an extension $\lambda \in T_{i-1}^k$ and such that for some $x < 2^{-2i-4}|\lambda|$ every extension ρ of λ in $\widehat{T}_i^{k,s}$ for any s has $\Phi_i^\rho(x) \uparrow$. In this case, set T_i^k to be the tree obtained by adding to T_i^{k-1} the downward closure of the string λ. Set ξ_i^k to be λ, and set $\sigma^k(i) = 1$, and proceed to the construction of T_{i+1}^k.

If no such leaf λ_0 exists, set $\xi_i^k = \xi_i^{k-1}$ and $T_i^k = T_i^{k-1}$, and set $\sigma^k(i) = 0$, and proceed to the construction of T_{i+1}^k.

Constructing the new tree T_k^k Finally, we define $\xi_k^k = \xi_{k-1}^k$, $\sigma^k(k) = -1$, and we let T_k^k be the downward closure of ξ_k^k.

This concludes the construction of the trees T_i^k for $i \leq k$.

5 Verification

It now remains to be seen that the construction carried out above will satisfy our requirements, namely, that for each i, the ξ_i^k converge to some ξ_i, and that the ξ_i are among the initial segments of some real X. Furthermore, we need $X \leq_T \emptyset'$ and $\dim_P(X) > 0$, but whenever Φ_i^X is a total reduction, we require $\dim_P(\Phi_i^X) < 1$.

Theorem 5.1 *For each i, there is some L such that when $L < k_1 < k_2$, we have $\xi_i^{k_1} = \xi_i^{k_2}$, $\sigma^{k_1}(i) = \sigma^{k_2}(i)$, equal to either 0 or 1, and there is some string $\rho \in T_{i-1}^k$ such that $|\rho| \geq i$, $\rho \preceq \xi_i^k$, and $K(\rho) \geq \frac{1}{4}|\rho| - 1$. In addition, we may choose L so that if $k > L$, then for $s \geq k$ we have $\widehat{T}_i^{k,s} = T_i^s$. Now define*

$$T_i = \bigcup_k T_i^k$$

and

$$\xi_i = \lim_k \xi_i^k.$$

Then T_i is a c.e. pruned clumpy tree, and there is a path X through T_i such that for each k, each ξ_j^k is an initial segment of X for each $j \leq k$. In particular, ξ_i is an initial segment of X.

Proof In proving this theorem, it will be useful to prove several other properties to hold throughout the induction. Therefore we will also show that at each stage k, we define $\xi_i^k \in T_i^k$ with the properties that $\xi_i^k \preceq \xi_i^{k+1}$, and either $\xi_{i-1}^k \preceq \xi_i^k$ or vice versa. In addition, $T_i^{k-1} \subseteq T_i^k$ and $T_i^k \subseteq T_{i-1}^k$. We will also see that whenever $A = A \cap \rho 2^{\leq|\rho|}$ is a pruned clump on T_i^k for $k > L$, and A's root is an initial segment of a path through T_i, A contains at least 2^{q_ρ} many leaves of $\tau 2^{\leq|\rho|}$, where q_ρ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i} 2^{-2j-4})|\rho|$.

Note that for the case $i = -1$, ξ_{-1}^k and T_{-1}^k are chosen specifically so that they meet the requirements of the theorem, as well as the additional properties mentioned above.

Also note that at each stage of the construction we preserve the requirement that $T_i^{k-1} \subseteq T_i^k$ and that $T_i^k \subseteq T_{i-1}^k$ for each $i < k$, and ensure that $T_k^k \subseteq T_{k-1}^k$ by choice, so that these conditions will hold of all the trees we build. In addition, at each stage of the construction we ensure that $\xi_j^k \preceq \xi_i^k$ or $\xi_i^k \preceq \xi_j^k$ for each $i, j < k$, and define $\xi_k^k = \xi_{k-1}^k$, so that comparability of the strings is preserved at every stage. Furthermore, we always set $\xi_i^{k-1} \preceq \xi_i^k$. But this means that if we let τ^k be the longest of the ξ_i^k at stage k, then we have $\tau^k \preceq \tau^{k+1}$ at each stage k, so it is clear that there is a path through all of the τ^k which lies in T_{-1}.

Now let $i \geq 0$, and fix L so large that the statements in the theorem and the additional hypotheses mentioned hold for each $j < i$ at stage L of the construction.

Lemma 5.2 *For sufficiently large M, whenever $k > M$, $\xi_i^k = \xi_i^M$ and $\sigma^k(i) = \sigma^M(i) \neq -1$. Furthermore, there is some string ρ with $|\rho| \geq i$, $\rho \preceq \xi_i^k$, and $K(\rho) > \frac{1}{4}|\rho| - 1$. In particular, $|\xi_i^k| \geq i$.*

In the case that $k > M$ and $\sigma^k(i) = 1$, there is some x such that each string $\rho \in T_{i-1}$ which extends ξ_i^k has $\Phi_i^\rho(x) \uparrow$.

On the other hand, if $k > M$ and $\sigma^k(i) = 0$, then for each string $\lambda \in T_i$ which is the root of a pruned clump on T_i with $\xi_i^k \preceq \lambda$, and each leaf ρ of that clump, there is a string $\widehat{\rho} \in T_i$ with $\rho \preceq \widehat{\rho}$ and $\Phi^{\hat{\rho}}(x) \downarrow$ for each $x < 2^{-2i-4}|\lambda|$. Furthermore, for any two extensions $\widehat{\rho_1}, \widehat{\rho_2}$ of λ on T_i, and each $x < 2^{-2i-4}$, we have $\Phi_i^{\hat{\rho}_1}(x) = \Phi_i^{\hat{\rho}_1}(x)$.

Proof By assumption, there is a path through T_{i-1} which has ξ_j as an initial segment for each $j < i$. Thus we may choose M_0 large enough to satisfy all of the hypotheses imposed for $j < i$, and such that there is a pruned clump A which is on each T_{i-1}^k for stages $k > M_0$, and whose root extends ξ_j for $j < i$, and is of length greater than i. Assume that for no $j < i$ do we have $\sigma^k(j) = -1$ for any $k > M_0$.

Now suppose that for some $k > M_0$, $\sigma^{k-1}(i) = -1$. Note that for each $j > i$ we also have $\sigma^k(j) = -1$, and $\xi_j^k = \xi_i^k$, because we only allow the least i with $\sigma^{-1}(i) = -1$ to make any changes at stage k. The construction of T_i^k proceeds by searching on T_i^{k-1} for a string ρ with $|\rho| \geq i$ and $K(\rho) > \frac{1}{4}|\rho| - 1$, and which is consistent with our choices ξ_j^k for $j < i$ and ξ_j^{k-1} for $i < j < k$. There is some string $\rho \in A$ with the property that $K(\rho) > \frac{1}{4}|\rho| - 1$ by Lemma 2.2 and the inductive hypothesis, and so we will set $\xi_i^k = \rho$ for some such ρ.

Because of our choice of M_0, we know that there is no stage $k > M_0$ at which $\sigma^k(j)$ changes for any $j < i$. Therefore once ξ_i^k has an initial segment satisfying the complexity condition discussed above, the only way that we can have a stage $k' > k$ for which $\xi_i^{k'-1} \neq \xi_i^{k'}$ is if $\sigma^{k'}(i) = 1$ but $\sigma^{k'-1}(i) = 0$. But in this case we choose $\xi_i^{k'}$ to be a string which extends $\xi_i^{k'-1}$, and thus inherits satisfaction of the complexity condition. Also note that if we never set $\sigma^{k_1}(i) = 1$ at any stage $k_1 > k$, then every time we add a pruned clump to $T_i^{k_1}$ with root λ, it is added in a single step, and we add extensions $\widehat{\rho}$ of each leaf ρ of the pruned clump so that for each $x < 2^{-2i-4}|\lambda|$, we have $\Phi_i^{\hat{\rho}}(x) \downarrow$, and so that for any two such extensions $\widehat{\rho_1}, \widehat{\rho_2}$ of λ on T_i, if $\Phi_i^{\hat{\rho}_1}(x) \downarrow$ and $\Phi_i^{\hat{\rho}_2}(x) \downarrow$ for some $x < 2^{-2i-4}$, we have $\Phi_i^{\hat{\rho}_1}(x) = \Phi_i^{\hat{\rho}_1}(x)$. This agreement of halting computations will obviously be preserved under extensions, and since the construction only ever extends strings which are already leaves, will hold of all extensions of λ in T_i.

Finally, note that if $\sigma^{k_1}(i) = 1$ at some stage $k_1 > M_0$, and we set τ to be the longest string in the set $\{\xi_{-1}^{k_1}, \xi_0^{k_1}, \xi_1^{k_1}, \ldots, \xi_{i-1}^{k_1}, \xi_i^{k_1-1}, \ldots, \xi_{k_1-1}^{k_1-1}\}$, then we know that there is some leaf λ_0 on $T_i^{k_1-1}$ which extends τ, and which has an extension $\lambda \in T_{i-1}^{k_1}$ such that for some $x < 2^{-2i-4}|\lambda|$, every extension ρ of λ in $\widehat{T}_{i-1}^{k_1,s}$ for any s has $\Phi_i^\rho(x)\uparrow$. In this case we have set $\xi_i^{k_1} = \lambda$, and $T_i^{k_1}$ is the tree obtained by adding the downward closure of λ to $T_i^{k'-1}$.

By the assumption on $\widehat{L}$, it follows from the inductive hypothesis on the trees $\widehat{T}_{i-1}^{k,s}$ that the strings which are in $\widehat{T}_{i-1}^{k,s}$ for some s are precisely those strings in T_{i-1}. From this we may therefore conclude that there is some x such that every string ρ on T_{i-1} which extends $\xi_i^{k'}$ now has the property that $\Phi_i^\rho(x)\uparrow$, as desired.

It clearly suffices to take $M > M_0$ to be some stage such that for $k > M$, $\xi_i^k = \xi_i^M$ and such that if $\sigma^k(i) = 1$ for any $k > M$, this is true of every $k > M$. □

Lemma 5.3 *For sufficiently large M, whenever $k > M$, and $s \geq k$, $T_i^s = \widehat{T}_i^{k,s}$. Thus T_i is the union of the $\widehat{T}_i^{k,s}$ at such stages.*

Proof Choose $M > L$ larger than the M of the previous lemma, so that ξ_j^k and $\sigma^k(j)$ are constant for all $j \leq i$ and $k > M$. Then note that by inductive assumption, for $s \geq k$ we have $\widehat{T}_{i-1}^{k,s} = T_{i-1}^s$. Fix some $k > M$. Notice that we have $\widehat{T}_i^{k,k} = T_i^k$ by definition. In addition, because $\sigma^s(j)$ and ξ_j^s are fixed for $j \leq i$ and $s \geq k$, it follows that the construction of T_i^s for $s \geq k$ either always proceeds via application of $\mathcal{S}$ or always by applications of $\mathcal{T}$, and that the algorithm which constructs $\widehat{T}_i^{k,s}$ will use the same algorithm, taking as input $\widehat{T}_i^{k,s-1}$ and $\widehat{T}_{i-1}^{k,s}$ rather than T_i^{s-1} and T_{i-1}^s. In either case, for each $s \geq k$ we know that $\widehat{T}_{i-1}^{k,s} = T_{i-1}^s$, and it will follow inductively that $T_i^s = \widehat{T}_i^{k,s}$. □

Lemma 5.4 *For sufficiently large M, whenever $k_0 > M$, for each leaf λ_0 of $T_i^{k_0}$ which is extended by some path through T_i, there is some $k > k_0$ and a pruned clump on T_i^k whose root λ is an extension of λ_0. Furthermore, this clump has at least 2^{q_λ} many leaves of $\lambda 2^{\leq|\lambda|}$, where q_λ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i} 2^{-2j-4})|\lambda|$.*

Proof Take M to be large enough to satisfy each of the previous lemmas, and $k_0 > M$.

Fix some leaf λ_0 of $T_i^{k_0}$ which is an initial segment of a path through T_i. Notice that this means that we must have $\xi_i^{k_0} \preceq \lambda_0$, since after stage k_0, we only ever add strings to T_i which extend $\xi_i^{k_0}$. Now T_{i-1} is a pruned clumpy tree and λ_0 lies on a path through it (since $T_i \subseteq T_{i-1}$), so we know that there is a pruned clump on T_{i-1} with a root extending λ_0, and that we may choose the root to be as long a string as we like (but the form of T_{-1} forces all pruned clumps to have roots whose lengths are powers of 4). Thus, for large enough k_1 we may note that λ_0 has an extension $\lambda \in T_{i-1}^{k_1}$ which is the root of a pruned clump on T_{i-1}, and such that $4^N = |\lambda| > 4|\lambda_0|$ for some $N > i + 2$. By assumption, the pruned clump on T_{i-1}

has at least 2^{q_λ} many leaves of $\lambda 2^{\leq|\lambda|}$, where q_λ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i-1} 2^{-2j-4})|\lambda|$.

Now suppose that we have $\sigma^k(i) = 0$; then we know that at no stage $k > k_0$ do we set $\sigma^k(i) = 1$. But this means that at each stage $k > k_0$ of the construction, we only ever add extensions to T_i^k if they are given by running $\mathcal{S}$ to search for them in T_{i-1}^k. Choose $k_1 > k$ large enough that λ_0 is extended by a pruned clump A with root $\lambda \succeq \lambda_0$ on $T_{i-1}^{k_1}$ whose length is of form $4^N > 4|\lambda_0|\cdot 2^{2i+4}$ for some $N > i+2$. Also assume that for each leaf on A we have some extension $\rho \in T_{i-1}^{k_1}$ with $\Phi_{i,k_1}^{\rho}(x) \downarrow$ for each $x < 2^{-2i-4}|\lambda|$ (if this is not true at any stage k_1, then we will never extend λ_0 in T_i, so it is not on a path through that tree, contrary to assumption). At the first such stage k_1 of the construction, the algorithm $\mathcal{S}$ will detect the extensions of λ, and will add a pruned clump to $T_i^{k_1}$.

The pruned clump B on $T_i^{k_1}$ with root λ is given as follows. The strings which are added to B are chosen from among those of the pruned clump A. For each leaf $\widehat{\rho}$ of B we will have $\Phi_i^{\hat{\rho}}(x) \downarrow$ whenever $x < 2^{-2i-4}|\lambda|$. Furthermore, any two such leaves will output the same $2^{-2i-4}|\lambda|$ many bits in this computation; we take B to have the maximum possible number of leaves extending λ and satisfying this condition. Note that there are $2^{2^{-2i-4}|\lambda|}$ binary strings of length $2^{-2i-4}|\lambda|$, and that the pruned clump A on $T_{i-1}^{k_1}$ with root λ has at least 2^{q_λ} many leaves, where q_λ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i-1} 2^{-2j-4})|\lambda|$. From this it follows by a simple count that B will have at least $2^{q'_\lambda}$ leaves, where q'_λ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i} 2^{-2j-4})|\lambda|$.

If $\sigma^k(i) = 1$, then there is a pruned $A \subseteq T_i^k$ with root $\lambda \succeq \lambda_0$, since at this stage all extensions of ξ_i^k on T_{i-1}^k are added to T_i^k. Notice that the number of leaves on this pruned clump is at least 2^{q_λ}, where q_λ is the least integer greater than or equal to $(1 - \sum_{j=-1}^{i-1} 2^{-2j-4})|\lambda|$, which is greater than $(1 - \sum_{j=-1}^{i} 2^{-2j-4})|\lambda|$. □

This concludes the proof of the inductive claims on the T_i.

Define $\xi_i = \lim_k \xi_i^k$ for each i. Note that we have $\xi_i \preceq \xi_j$ or vice versa for every i and j, because this is true at each stage of the construction. Note that each tree T_i contains ξ_j for every j, since $\xi_j^k \in T_i^k$ whenever $j \leq k$. But we have $|\xi_j| \geq j$ for every j, and so it follows that there is a unique path X through T_i which has each of the ξ_j as initial segments. □

The real X with initial segments given by the ξ_j is the real which we desired to construct.

Lemma 5.5 *With X as constructed,* $\dim_P(X) \geq \frac{1}{4}$.

Proof For each j, there is some initial segment ρ of X with $K(\rho) > \frac{1}{4}|\rho| - 1$ and $|\rho| \geq j$. Therefore it follows that $\dim_P(X) = \limsup_n \frac{K(X \upharpoonright n)}{n} \geq \frac{1}{4}$. □

Now we must show that the conditions given on the trees are sufficient to guarantee that if Y is a real computed by X, then $\dim_P(Y) < 1$.

6 Combinatorial Estimate

Note that if $\sigma^k(i)$ has limit 1, then for some x, every string ρ on T_i has $\Phi_i^\rho(x)\uparrow$, and so in particular Φ_i cannot compute any reals from X. Therefore to show that for each i such that Φ_i^X computes a real, $\dim_P(\Phi_i^X) < 1$, we need only consider i such that $\sigma^k(i)$ has limit 0.

Thus, throughout this section, suppose that we are in the case that $\sigma^k(i)$ has limit 0. In this case, every time we added a pruned clump to T_i with a root λ extending ξ_i, we made sure that for each leaf ρ of that pruned clump, we also added some extension $\widehat{\rho}$ of ρ with the property that $\Phi_i^{\hat{\rho}}(x)\downarrow$ for each $x < 2^{-2i-4}|\lambda|$, and furthermore, for any extensions $\widehat{\rho_1}, \widehat{\rho_2}$ of λ such that for some $x < 2^{-2i-4}|\lambda|$, $\Phi_i^{\hat{\rho}_1}(x)\downarrow$ and $\Phi_i^{\hat{\rho}_2}(x)\downarrow$, we have $\Phi_i^{\hat{\rho}_1}(x) = \Phi_i^{\hat{\rho}_2}(x)$.

The goal of this section is to provide an upper bound on the Kolmogorov complexity of the strings in $\Phi_i^{T_i}$ which are initial segments of Φ_i^X, and thus to show that this path has effective packing dimension less than 1. We will do so by defining two prefix-free machines M_1 and M_2 which will provide descriptions of strings in $\Phi_i^{T_i}$, using the construction of T_i to provide short descriptions.

The computation presented here is perhaps less elegant than that presented by Conidis [3], but more general. Its directness is intended to offer some intuition as to why the complexities of initial segments of paths through $\Phi_i^{T_i}$ are bounded away from 1, as well as indicating roughly where the strings of (comparatively) high complexity might be found on those paths.

Throughout what follows, we will regard T_i as a computably enumerable set of strings, with the enumeration given by

$$T_i = \bigcup_{s\geq k} T_i^k,$$

where k is some stage chosen to be sufficiently large that $T_i^s = \widehat{T}_i^{k,s}$ for each $s \geq k$.

In order to have a record of what order clumps were added to the tree T_i, number the pruned clumps on T_i^k via a computable numbering N as follows. If A is the unique pruned clump on T_i with root $\rho \succ \xi_i$, and there is no pruned clump on T_i with root $\widehat{\rho}$ such that $\xi_i \prec \widehat{\rho} \prec \rho$, then set $N(A) = 1$. Supposing that we have defined the pruned clumps A for which $N(A) = j$, define $N(B) = j+1$ for a pruned clump B exactly when the root ρ of B extends the root ρ_0 of some pruned clump A with $N(A) = j$, but such that for no pruned clump C with root $\widehat{\rho}$ do we have $\rho_0 \prec \widehat{\rho} \prec \rho$. It should be noted that if A is a pruned clump on T_i and $N(A) = j$, and we are given some leaf λ on A such that some pruned clump on T_i has a root extending λ, it is possible to computably identify the unique pruned clump B with $N(B) = j+1$ and whose root extends λ. This is because the enumeration of T_i will add all of B at some stage, and this will be the first time it adds any extension of λ.

Definition 6.1 Let A_j be a pruned clump on T_i for $1 \leq j \leq n$, and for each j let ρ_j be the root of A_j. Suppose that we have $\rho_j \prec \rho_{j+1}$ for each $j < n$, and that $N(A_j) = j$ for each j. Then we call $A_1, A_2, \ldots, A_j$ a *sequence of adjacent pruned clumps*.

The idea is that if we have a string $\tau \in \Phi_i^{T_i}$ which is an extension of ξ_i, then for some string $\rho \in T_i$, we have $\Phi_i^\rho = \tau$. We will let the sequence $A_1, A_2, \ldots, A_n$ of adjacent pruned clumps consist of the pruned clumps on T_i whose roots ρ_i have $\rho_i \preceq \tau$. Notice that each pruned clump on T_i with a root of length l has at most 2^l leaves, and so each can be identified with a string of length l by assigning those strings according to the lexicographical order of the leaves. This assignment is computable, since all of the leaves on such a pruned clump are added simultaneously at some stage of the enumeration of T_i.

The first of the two prefix-free machines, M_1, will be used to compute those initial segments τ of the real Φ_i^X which have the property that

$$3(1 + 2^{-2i-4})|\rho| \leq |\tau| \leq 2^{-2i-4}|\rho|,$$

where ρ is a root of a pruned clump A on T_i, and the root $\widehat{\rho}$ of a pruned clump B with $N(B) = N(A) + 1$ extends ρ; in this case, there are extensions $\widehat{\lambda}$ of $\widehat{\rho}$ on T_i that have the property that $\Phi_i^{\hat{\lambda}}(x) = \tau(x)$ for each $x < |\tau|$. In this case, the computation used will find some extension of $\widehat{\rho}$ on T_i which gives a halting computation at least as long as τ, and then to note that extensions of $\widehat{\rho}$ giving such a computation all give the *same* computation, so that τ has been computed. In this case, we will have used approximately $\frac{2}{3}|\tau|$ many bits in the computation.

More specifically, M_1, on input of a binary string σ, attempts to decompose σ into the form $0^n 1 \theta_1 \theta_2 \cdots \theta_n 0^l 1 \mu$ as follows:

Step 1 The machine M_1 takes $|\theta_1|$ to be the length of the root of the unique pruned clump on T_i with $N(A) = 1$, and sets $A_1 = A$, unless σ is too short to parse in this way, in which case M_1 does not halt. Suppose that $k < n$, and M_1 has computed $\theta_1, \ldots, \theta_k$, and a sequence of adjacent pruned clumps $A_1, \ldots, A_k$ such that for $j \leq k$, A_k has root ρ_k with $|\rho_k| = |\theta_k|$. Then M_1 interprets θ_k as corresponding to some $r \in \omega$ with $r < 2^{|\theta_k|}$ as a binary expansion (possibly with some leading zeros), and searches for a pruned clump A_{k+1} on T_i which extends the rth leaf of A_k, and such that $N(A_{k+1}) = k + 1$. Note that if such a pruned clump exists, it is unique. If no such pruned clump exists, or if A_k does not have r leaves, M_1 does not halt.

Otherwise, letting ρ_{k+1} be the root of A_{k+1}, M_1 will set θ_{k+1} to be the $|\rho_{k+1}|$ bits of σ which follow those we have labeled θ_k. If the string σ does not have another $|\rho_{k+1}|$ many bits, then M_1 does not halt.

Step 2 Once M_1 has found $\theta_1, \ldots, \theta_n$ and $A_1, \ldots, A_n$, it then identifies the number l for which the bits immediately following θ_n are of the form $0^l 1$, and then, if there are precisely l bits following those in the string σ, interprets those l bits as the expansion μ of a binary number $m < 2^l$ (if the number of bits in μ is not correct, M_1 does not halt).

At this point, M_1 finds the string $\rho \in T_i$ which is the leaf on A_n whose lexicographic position is given by θ_n (we interpret the position as an n-bit binary number, possibly with some leading zeros). M_1 then searches for a pruned clump A_{n+1} on T_i with $N(A_{n+1}) = n + 1$ and whose root ρ_{n+1} satisfies $\rho_{n+1} \succeq \rho$. If no such pruned clump is found, M_1 will not halt. If such a pruned clump is found, note that there is some extension $\lambda \succeq \rho_{n+1}$ on T_i such that $\Phi_i^{\lambda}(x) \downarrow$ for each $x < 2^{-2i-4}|\rho_{n+1}|$, and that all strings $\widehat{\lambda}$ which extend ρ_{n+1} have the property that if $\Phi_i^{\hat{\lambda}}(x) \downarrow$ for some $x < 2^{-2i-4}$, then $\Phi_i^{\hat{\lambda}}(x) = \Phi_i^{\lambda}(x)$. This means that we can uniformly compute an unambiguous value $\Phi_i^\lambda(x) \upharpoonright 2^{-2i-4}|\rho_{n+1}|$ which is equal for all such λ. At this point, M_1 checks whether $m < 2^{-2i-4}|\rho_{n+1}|$. If not, then M_1 will not halt.

Finally, we are ready to give the output of M_1. Assuming that all of the computations given above are carried out correctly, and M_1 has not yet been stated not to halt, we set $M_1(\sigma) = \Phi_i^\lambda(x) \upharpoonright 2^{-2i-4}m$, with λ as in the previous paragraph.

Note that M_1 is a prefix-free machine, since any prefix of a string on which M_1 halts will at some point not have the correct syntactical form and will be rejected.

In what follows, the next two lemmas will prove vital in providing the estimates we require.

Lemma 6.2 *If $A_1, \dots, A_{n+1}$ is a sequence of pruned clumps, where A_j's root is ρ_j for each $j \leq n+1$, then for each $j \leq n$ we have $|\rho_{j+1}| \geq 4 \cdot 2^{2i+4}|\rho_j|$.*

Proof The construction of T_i ensures that if $\rho_j \preceq \rho_{j+1}$ and $N(A_j) = j$, and $N(A_{j+1}) = j+1$, then when we added the clump A_j to T_i, we added some extension λ of the leaf of A_j which ρ_{j+1} extends, by applying the algorithm $\mathcal{S}$. Because of the way this algorithm operates, we must then have $|\rho_{j+1}| \geq 4 \cdot 2^{2i+4}|\lambda| > 4 \cdot 2^{2i+4}|\rho_j|$. □

Lemma 6.3 *Let τ be a string in $\Phi_i^{T_i}$ that is an initial segment of the path Φ_i^X through $\Phi_i^{T_i}$, and assume that $A_1, \dots, A_{n+1}$ is a sequence of adjacent pruned clumps with ρ_j the root of A_j, that some extension ρ of ρ_{n+1} has $\Phi_i^\rho(x) = \tau(x)$ for each $x < |\tau|$, and that $2^{-2i-4}|\rho_n| < |\tau| \leq 2^{-2i-4}|\rho_{n+1}|$. Then we have*

$$K(\tau) \leq n + \sum_{j=1}^{n} |\rho_j| + 2\log_2\big(|\tau|\big) + C$$

for some C, and in particular if $|\tau| \geq 3(1 + 2^{-2i-4})|\rho_n|$, then we have

$$K(\tau) \leq \frac{2}{3}|\tau| + D\big(\log_2\big(|\tau|\big)\big)$$

for some D.

Proof The first result follows immediately. Let σ be given by $0^n 1\theta_1\theta_2 \cdots \theta_n 0^l 1\mu$, where $|\theta_j| = |\rho_j|$, and θ_j tells us which leaf of the pruned clump with root ρ_j has the root of A_{j+1} on it, and μ is the binary expansion for $|\tau|$, of length l. Then we have $M_1(\sigma) = \tau$.

In the case where we have $|\tau| \geq 3(1 + 2^{-2i-4})|\rho_n|$, we use the fact that the construction of T_i ensures that $|\rho_{j+1}| \geq 4 \cdot 2^{2i+4}|\rho_j|$ for each j; therefore we have

$$\begin{aligned}
\sum_{j=1}^{n} |\rho_j| &\leq \sum_{j=1}^{n-1} \Big(\frac{2^{-2i-4}}{4}\Big)^{n-j} |\rho_n| + |\rho_n| \\
&\leq \frac{3}{2}|\rho_n| \\
&\leq \frac{2}{3}|\tau|.
\end{aligned}$$

But then we have $K(\tau) \leq \frac{2}{3}|\tau| + n + 2\log_2|\tau| + C$, and noting that we have $|\tau| \geq |\rho_n| \geq 4^n$, it follows that $n \leq \log_2(|\tau|)$, and the result follows. □

Now I give a second prefix-free machine M_2 which will bound the complexity of other strings on $\Phi_i^{T_i}$. This machine will be used to compute initial segments of Φ_i^X in the cases which M_1 was unable to provide adequate bounds for.

The machine M_2 proceeds as follows. Given a string σ, M_2 tries to parse σ to be of the form $0^n 1\theta_1\theta_2\cdots\theta_n 0^l 1\mu 01\nu$ in a similar manner to M_1. It first mimics the behavior of M_1's step 1, to check whether the initial segment of the string is of the form $0^n 1\theta_1\theta_2\cdots\theta_n$, where the θ_j correspond to a sequence of adjacent pruned clumps, and to specify which leaf of A_j is to be extended to find the root of A_{j+1}. In this way it is either able to compute the root of the pruned clump A_{n+1}, or does not halt.

After this, M_2 checks whether μ is of length $2l$, and that for each $j < l$, the $2j$th and $(2j+1)$-st bits of μ are equal. If not, M_2 will not halt. If the property does hold of μ, M_2 then deletes every second bit of μ to obtain a string of length l; it interprets this string as a binary expansion of a number L, and then checks whether L is the length of the string ν. If so, then M_2 searches for an extension λ of the root ρ_{n+1} of A_{n+1} such that $\Phi_i^\lambda(x)\downarrow$ for every $x < 2^{-2i-4}|\rho_{n+1}|$. It then takes $\widehat{\tau}$ to be the string such that $|\widehat{\tau}| = 2^{-2i-4}|\rho_{n+1}|$ and $\Phi_i^\lambda(x) = \widehat{\tau}(x)$ for each $x < 2^{-2i-4}|\widehat{\tau}|$.

Finally, we have $M_2(\sigma) = \widehat{\tau}\nu$, that is, the concatenation of the two strings.

Lemma 6.4 *Let τ be a string in $\Phi_i^{T_i}$ that is an initial segment of the path Φ_i^X through $\Phi_i^{T_i}$, and assume that $A_1,\dots,A_{n+1}$ is a sequence of adjacent pruned clumps with ρ_j the root of A_j, that some extension ρ of ρ_{n+1} has $\Phi_i^\rho(x) = \tau(x)$ for each $x < |\tau|$, and that $2^{-2i-4}|\rho_n| < |\tau| \le 2^{-2i-4}|\rho_{n+1}|$. Then we have*

$$K(\tau) \le n + \sum_{j=1}^{n-1}|\rho_j| + 3\log_2\big(|\tau|\big) + |\tau| - 2^{-2i-4}|\rho_n| + C$$

for some C, and in particular if $|\tau| \le 3(1+2^{-2i-4})|\rho_n|$, then we have

$$K(\tau) \le \Big(1 - \frac{2^{-2i-4}}{6(1+2^{-2i-4})}\Big)|\tau| + D\log_2\big(|\tau|\big)$$

for some D.

Proof Once again, the first result follows very easily. Let σ be given by $0^n 1\theta_1\theta_2\cdots\theta_{n-1}0^l 1\mu\nu$, where θ_j tells us which leaf of the pruned clump with root ρ_j has the root of A_{j+1} on it, and μ is the doubled binary expansion for $|\nu|$, which is of length l. Then we have $M_2(\sigma) = \tau$. The $3\log_2(|\tau|)$ term comes from the fact that $|\mu| \le \log_2(\tau)$ and so $|0^l 1\mu| \le 3\log_2(\tau) + 1$, whereas $|\tau| - 2^{-2i-4}|\rho_n|$ is the length of ν.

Suppose that we have $|\tau| \le 3(1+2^{-2i-4})|\rho_n|$. Note that we have

$$\begin{aligned}\sum_{j=1}^{n-1}|\rho_j| &\le \sum_{j=1}^{n-1}\Big(\frac{2^{-2i-4}}{4}\Big)^{n-j}|\rho_n|\\ &\le \frac{2^{-2i-4}}{2}|\rho_n|.\end{aligned}$$

Now we once again bound all of the smaller terms by a single term which is logarithmic in size, and apply our assumption on $|\tau|$:

$$\begin{aligned}K(\tau) &\leq \frac{2^{-2i-4}}{2}|\rho_n| + |\tau| - 2^{-2i-4}|\rho_n| + D\log_2\big(|\tau|\big)\\ &= |\tau| - \frac{2^{-2i-4}}{2}|\rho_n| + D\log_2(\tau)\\ &\leq |\tau| - \frac{2^{-2i-4}}{6(1+2^{-2i-4})}|\tau| + D\log_2\big(|\tau|\big)\\ &= \Big(1 - \frac{2^{-2i-4}}{6(1+2^{-2i-4})}\Big)|\tau| + D\log_2\big(|\tau|\big),\end{aligned}$$

which provides the bound we sought. □

Now for each string τ which lies on the path Φ_i^X through $\Phi_i^{T_i}$, we have a sequence of pruned clumps $A_1, \ldots, A_{n+1}$ with roots ρ_j such that some extension ρ of ρ_{n+1} has $\Phi_i^\rho(x) = \tau(x)$ for every x, and that if τ is sufficiently long, we may choose the sequence so that $2^{-2i-4}|\rho_n| < |\tau| \leq 2^{-2i-4}|\rho_{n+1}|$, simply by choosing a sequence of adjacent pruned clumps which is of suitable length. This is sufficient for our needs regarding the reals computed by X.

Lemma 6.5 *For some D, all sufficiently long initial segments τ of Φ_i^X have the property that*

$$K(\tau) \leq \Big(1 - \frac{2^{-2i-4}}{6(1+2^{-2i-4})}\Big)|\tau| + D\log_2\big(|\tau|\big).$$

We have now completed all of the work which is required to prove the main theorem (Theorem 1.2).

Proof of Theorem 1.2 If X is the real defined by taking the ξ_i as initial segments, then $X \leq_T \emptyset'$ by construction. In addition, we have $\dim_P(X) \geq \frac{1}{4}$, as seen in Lemma 5.5. Furthermore, for any i such that $\Phi_i^X = Y$ is a total reduction, we have $\dim_P(Y) < 1$. □

Finally, we consider a result on extraction of Kolmogorov complexity.

Theorem 6.6 ([7, Theorem 5.2], as stated by [2, Theorem 2.5]) *For any $X \in 2^\omega$ with $\dim_P(X) > 0$ and any $\epsilon > 0$, there is some $Z \in 2^\omega$ such that $Z \equiv_T X$ and $\dim_P(Z) \geq 1 - \epsilon$.*

This result, applied directly to the Turing degree of X, yields Corollary 1.3.

7 Future Directions

Many results which prove the existence of real numbers possessing properties expressed in terms of Kolmogorov complexity, effective packing dimension, and effective Hausdorff dimension are able to be effectivized to show that such reals exist below any array noncomputable degree (see, e.g., papers of Downey and Greenberg [4], Downey and Ng [6], and Kummer [9]). This common trend suggests that the obvious analogue may hold in the current case.

Question 7.1 Given an array noncomputable degree $\mathbf{a}$, is there a real $X \leq_T \mathbf{a}$ which has nonzero effective packing dimension, but which cannot compute any real of effective packing dimension 1?

Given the suggestive pattern noted above, it seems reasonable to suppose that the answer is very likely "yes." Indeed, the author feels that the proof is likely to proceed somewhat as follows. Noting that the only oracle questions which we are asking are about either the prefix-free Kolmogorov complexity of a string, or asking whether some string forces divergence of an oracle machine, rewrite the list of requirements for the construction so that the oracle questions asked of $\emptyset'$ are given as a uniformly computable list of questions which are wtt-computable by $\emptyset'$. But we know that any array noncomputable real will answer infinitely many of these oracle questions correctly, and so such a real should suffice to find strings of high complexity if given a sequence of pruned clumps, and likewise to find strings which force divergence of computations.

It should be noted that if a real X has nonzero effective packing dimension, but cannot compute any real of effective packing dimension 1, then X is necessarily itself array computable, as an immediate corollary of a result of Downey and Greenberg [4, Theorem 1.5].

Theorem 7.2 ([4]) *Every array noncomputable degree* $\mathbf{a}$ *computes a set* A *with effective packing dimension* 1.

From this it is clear that the array noncomputable degrees cannot be the set of degrees which compute a real X as above. However using the work of [9, Theorem 2.2] Kummer and the above result, it follows that among the c.e. degrees, array computability may be characterized as follows.

Theorem 7.3 ([5, Theorem 1.5]) *A c.e. degree is array noncomputable if and only if it computes a real* Y *with* $\dim_P(Y) > 0$.

This implies that no degree whose members are computed by an array computable c.e. degree can have nonzero effective packing dimension. From this we see that the only c.e. degrees below which it is possible to carry out a construction of a degree with the properties set out in this paper are the array noncomputable ones. As suggested above, it seems likely that this is precisely the set of c.e. degrees below which such a construction can occur. This conjecture will be explored in a future paper.

References

[1] Athreya, K. B., J. M. Hitchcock, J. H. Lutz, and E. Mayordomo, "Effective strong dimension in algorithmic information and computational complexity," *SIAM Journal on Computing*, vol. 37 (2007), pp. 671–705. Zbl 1144.68029. MR 2341912. DOI 10.1137/S0097539703446912. 75

[2] Bienvenu, L., D. Doty, and F. Stephan, "Constructive dimension and Turing degrees," *Theory of Computing Systems*, vol. 45 (2009), pp. 740–55. Zbl 1183.68281. MR 2529744. DOI 10.1007/s00224-009-9170-1. 74, 90

[3] Conidis, C. J., "A real of strictly positive effective packing dimension that does not compute a real of effective packing dimension one," *Journal of Symbolic Logic*, vol. 77 (2012), pp. 447–74. MR 2963016. 74, 75, 77, 78, 86

[4] Downey, R., and N. Greenberg, "Turing degrees of reals of positive effective packing dimension," *Information Processing Letters*, vol. 108 (2008), pp. 298–303. Zbl 1191.68304. MR 2456604. DOI 10.1016/j.ipl.2008.05.028. 73, 90, 91

[5] Downey, R. G., and D. R. Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer, New York, 2010. MR 2732288. DOI 10.1007/978-0-387-68441-3. 73, 74, 91

[6] Downey, R., and K. M. Ng, "Effective packing dimension and traceability," *Notre Dame Journal of Formal Logic*, vol. 51 (2010), pp. 279–90. Zbl 1204.03042. MR 2667937. DOI 10.1215/00294527-2010-017. 73, 90

[7] Fortnow, L., J. M. Hitchcock, A. Pavan, N. V. Vinodchandran, and F. Wang, "Extracting Kolmogorov complexity with applications to dimension zero-one laws," pp. 335–45 in *Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming, Part 1*, vol. 4051 of *Lecture Notes in Computational Science*, Springer, Berlin, 2006. MR 2305538. DOI 10.1007/11786986_30. 74, 90

[8] Greenberg, N., and J. S. Miller, "Diagonally non-recursive functions and effective Hausdorff dimension," *Bulletin of the London Mathematical Society*, vol. 43 (2011), pp. 636–54. MR 2820150. DOI 10.1112/blms/bdr003. 74

[9] Kummer, M., "Kolmogorov complexity and instance complexity of recursively enumerable sets," *SIAM Journal on Computing*, vol. 25 (1996), pp. 1123–43. Zbl 0859.03015. MR 1417892. DOI 10.1137/S0097539794268789. 90, 91

[10] Lutz, J. H., "The dimensions of individual strings and sequences," *Information and Computation*, vol. 187 (2003), pp. 49–79. Zbl 1090.68053. MR 2018734. DOI 10.1016/S0890-5401(03)00187-1. 74, 75

[11] Miller, J. S., "Extracting information is hard: A Turing degree of non-integral effective Hausdorff dimension," *Advances in Mathematics*, vol. 226 (2011), pp. 373–84. MR 2735764. DOI 10.1016/j.aim.2010.06.024. 74

[12] Sacks, G. E., "On the degrees less than $\mathbf{0}'$," *Annals of Mathematics (2)*, vol. 77 (1963), pp. 211–31. Zbl 0118.25104. MR 0146078. 74

[13] Soare, R. I., *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*, in *Perspectives in Mathematical Logic*, Springer, Berlin, 1987. MR 0882921. DOI 10.1007/978-3-662-02460-7. 76

[14] Soare, R. I., *Computability Theory and Applications: The Art of Classical Computability*, Springer, Berlin, 2016. Zbl pre06236171. 76

[15] Spector, C., "On degrees of recursive unsolvability," *Annals of Mathematics (2)*, vol. 64 (1956), pp. 581–92. Zbl 0074.01302. MR 0082457. 74

[16] Yates, C., "Initial segments of the degrees of unsolvability, I: A survey," pp. 63–83, in *Mathematical Logic and Foundations of Set Theory (Jerusalem, 1968)*, edited by Y. Bar-Hillel, vol. 59 of *Studies in Logic and the Foundations of Mathematics*, North Holland, Amsterdam, 1970. MR 0269505. 75

Acknowledgments

The author would like to thank Rod Downey for suggesting the problem and for many useful conversation as well as Chris Conidis for his advice.

This article is work toward a Ph.D. in Mathematics at the University of Chicago under Professors Robert Soare and Denis Hirschfeldt, to whom thanks are also extended.

Department of Pure Mathematics
University of Waterloo
Waterloo, ON
Canada
j9tephenson@uwaterloo.ca
http://uwaterloo.ca/pure-mathematics/about/people/j9stephe