PHYSICAL REVIEW D **96,** 103512 (2017)

Fast optimization algorithms and the cosmological constant

Ning Bao,[1] Raphael Bousso,[2,3] Stephen Jordan,[4,5] and Brad Lackey[4,6,7]

[1]*Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics, California Institute of Technology, Pasadena, California 91125, USA*

[2]*Center for Theoretical Physics and Department of Physics, University of California, Berkeley, California 94720, USA*

[3]*Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA*

[4]*Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, Maryland 20742, USA*

[5]*National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA*

[6]*Departments of Computer Science and Mathematics, University of Maryland, College Park, Maryland 20742, USA*

[7]*Mathematics Research Group, National Security Agency, Ft. G. G. Meade, Maryland 20755, USA*

(Received 27 July 2017; published 13 November 2017)

Denef and Douglas have observed that in certain landscape models the problem of finding small values of the cosmological constant is a large instance of a problem that is hard for the complexity class NP (Nondeterministic Polynomial-time). The number of elementary operations (quantum gates) needed to solve this problem by brute force search exceeds the estimated computational capacity of the observable Universe. Here we describe a way out of this puzzling circumstance: despite being NP-hard, the problem of finding a small cosmological constant can be attacked by more sophisticated algorithms whose performance vastly exceeds brute force search. In fact, in some parameter regimes the average-case complexity is polynomial. We demonstrate this by explicitly finding a cosmological constant of order 10^{-120} in a randomly generated 10^9-dimensional Arkani-Hamed–Dimopoulos–Kachru landscape.

DOI: 10.1103/PhysRevD.96.103512

I. INTRODUCTION AND SUMMARY

A. Cosmological constant problem and the landscape

According to the Standard Model of particle physics, the energy density of the vacuum receives multiple contributions whose order of magnitude vastly exceeds the observed value [1–3]

$$\Lambda \approx 1.5 \times 10^{-123} M_P^4. \tag{1}$$

[Below we will use units where the Planck mass is unity, $1 \equiv M_P = (\hbar c/G)^{1/2} \approx 1.2 \times 10^{19}$ GeV]. Both perturbative and nonperturbative processes contribute, such as vacuum fluctuations of all fields, and electroweak symmetry breaking. The excess is by a factor of at least 10^{60} assuming a new symmetry at a TeV (so far not found). It could be as large as 10^{122} with a Planck-scale cutoff. The observed small value of Λ implies that the various contributions must cancel against one another, or against further unknown contributions which must be at least as large, with a relative precision of at least 10^{-60} and perhaps 10^{-122}.

Consistency with well-established cosmological history severely constrains large classes of approaches to this problem. For example, it is not possible for the Universe to dynamically select the "correct" vacuum energy at early times. Only gravity couples to the absolute energy, and gravity sees the total stress tensor. At the time of big bang nucleosynthesis, characteristic energy densities were of order 10^{-88}. This is more than 30 orders of magnitude greater than the observed value that would have to be targeted by a putative adjustment mechanism. Attempts to desensitize general relativity to the energy in vacuum fluctuations run into conflict with tests of the equivalence principle. These and other obstructions to nonanthropic approaches are discussed in [4,5].

In a landscape model, a small cosmological constant is selected by correlation with the location of observers. The Universe can form large regions with many different possible values of Λ. This is most natural in a theory with extra dimensions, such as string theory. One finds that there are generically exponentially many ways of constructing a "vacuum," i.e., a compactification to three large spatial dimensions. If the vacuum energy Λ is, say, a random number between -1 and 1, but there are $\mathcal{N} \gg 10^{122}$ different vacua, it is likely that a small fraction but large number $10^{-122}\mathcal{N}$ of vacua have small enough Λ to be consistent with observation. Moreover, a great variety of vacua are naturally produced by inflationary dynamics in the early Universe. In specific models, the distribution of Λ is not random. The above approach works as long as the spectrum of Λ is sufficiently dense near 0. Consistency with standard cosmological history is achieved if the potential landscape is multidimensional, with neighboring vacua generically having very different energies [6].

Typical spacetime regions would still have $\Lambda \sim O(1)$, of course. But in such regions any worldline has an event

horizon of order the Planck area, and so contain only a few bits of causally connected information [7,8]. Complex structures such as observers necessarily find themselves in a highly atypical region that allows for a larger cosmological horizon with area (and hence, maximum entropy) of order Λ^{-1}. (The origin of the particular scale 10^{-122} is not explained by this qualitative argument. See [9] for an argument that assumes galaxies are needed or [10] for a more robust argument).

B. Computational complexity

In 2007, Denef and Douglas brought a complexity theoretic perspective to the cosmological constant problem [11]. In particular, they pointed out that, in some formulations, the problem of finding a vacuum with cosmological constant compatible with observation is a large instance of a NP-hard problem. Specifically, two simplified models were considered in [11]: a version of the Arkani-Hamed–Dimopoulos–Kachru (ADK) model [12] and the Bousso-Polchinski (BP) model [6]. Here we focus on the ADK model, which is the more simplified of the two, as it is sufficient to capture the essential features that we wish to address.

In the ADK model, the cosmological constant is obtained by summing the energy contributions from a large number of fields, each of which is subject to a double-well potential. We assume the vacuum energy contributed by either of the two minima of each field to be a random number with mean zero[1] and standard deviation of order 1 in Planck units. (Thus it can be positive or negative.) Given n such fields there are correspondingly $\mathcal{N} = 2^n$ metastable vacua, specified by an n-bit string $f(j) \in \{0, 1\}$, $j = 1, \ldots, n$. The cosmological constant in any vacuum is given by

$$\Lambda[f(j)] = \sum_{j=1}^{n} E^{(j)}_{f(j)}, \tag{2}$$

where $E^{(j)}_0$ and $E^{(j)}_1$ are the two possible vacuum energies contributed by the jth field.

If our Universe were described by this model, then with appropriate technology, there would be no obstruction in principle to measuring each of the n fields directly, and thus determining which of its two vacua it occupies. This requires only n measurements. Thus, we can in principle identify which vacuum we live in, among all the vacua in the ADK model. A similar argument applies to the BP model: given good enough technology, one would simply measure the fluxes on topological cycles in the extra dimensions. We could probe each field experimentally and read off the bit string $f(j)$.

[1]This assumption differs from the model mainly studied by ADK, but it is adequate for our analysis.

Denef and Douglas consider a different task: suppose we are given only the total value of the cosmological constant $\sim 10^{-122}$ (for example from observation), but not the vacuum configuration $f(j)$ of the n fields. We wish to identify a vacuum in the ADK model compatible with this value. Then we would have to sift through the 2^n allowed vacua to find a combination of positive and negative numbers, each of order 1, that add up to 10^{-122}. Such combinations clearly constitute a small fraction of all the 2^n vacua. However, in simple statistical models, e.g., where $E^{(1)}_0, E^{(1)}_1, \ldots, E^{(n)}_0, E^{(n)}_1$ are each independently drawn uniformly at random from $[-1, 1]$, such combinations will exist with high probability provided $\sqrt{n}2^{-n} \lesssim 10^{-122}$ [13], i.e., $n \gtrsim 407$. Furthermore, for n larger than this, the number of vacua with $\Lambda \leq 10^{-122}$ will be roughly $10^{-122} \times 2^n/\sqrt{n}$ [14,15].

In [11] it was pointed out that the problem of finding such vacua in the ADK model is a variant of the number partitioning problem, which is NP-complete. Consequently, under the widely held complexity-theoretic assumption that $P \neq NP$, no classical algorithm can solve worst-case instances of this problem in time scaling polynomially with n. Furthermore, under the stronger but also widely held assumption that $NP \not\subseteq BQP$, no quantum algorithm can solve worst-case instances of this problem in polynomial time either.

The physical significance of the Denef-Douglas observation is not immediately clear. Here, we posit that its significance lies in the contrast between the NP-complete hardness of finding a vacuum with small Λ by studying the theory, on the one hand; and on the other hand, the ease with which we can read off a solution to this problem (our own vacuum), by measuring the n bits directly as discussed above. This implies that we get to read off the answer to an instance of a NP-hard problem that nature has already solved for us. And we get to do this for anthropic reasons: complex structures exist only in regions with $\Lambda \ll 1$. Our mere status as observers gives us immediate access to the solution of a hard problem. How is this possible?

It is instructive to consider the cosmological dynamics that had to solve the "hard" problem and produce the small-Λ region we occupy. There are two valid and largely equivalent [16] viewpoints, global and local. In the global viewpoint, the Universe is exponentially expanding and constantly producing new regions. In this case gravity supplies exponential resources for solving the hard problem. No one can observe the whole Universe, because regions are shielded from one another by event horizons. But observers necessarily find themselves in the regions where the problem has been solved.

In the local viewpoint, one considers the different decay chains through the landscape that might be realized in a *single* causally connected region (causal patch). The patch decoheres rapidly every time a vacuum transition takes place. This trades the multiverse for "many worlds" [17].

Observers find themselves in a branch of the decay chain that produced a vacuum with small Λ. The situation is comparable to solving a hard problem by sitting down in front of a robot that points a gun at you. The robot takes one random guess (generated by some quantum measurement) and secretly checks it in polynomial time. If the guess solves the problem, the robot tells you the solution, but if it fails, it shoots you. Necessarily, if you survive, you will have gained the solution very quickly.[2]

We do not claim that from either of those viewpoints, our easy access to a solution of a hard problem constitutes a logical contradiction. Yet, the ability to utilize exponential unobservable resources or an exponentially large branching tree of decoherent histories would be a surprising and perhaps troubling circumstance. Therefore, in this paper, we will posit a *computational censorship hypothesis*: by physical measurements we must not be accessing the solution to a hard problem, i.e., a problem so hard that it could not have been solved by the physical resources in the observable Universe.

By "resources," we mean the number of elementary gates in a computation. There is some ambiguity about how to quantify an upper bound on this for the observable Universe. Possible candidates include (in natural units) the Einstein-Hilbert-matter action [20]; the energy of the Universe times its age [21]; the maximum entropy of the visible Universe [8,22] or of any universe with the observed value of Λ [23] (which is given by the horizon area of empty de Sitter space [7]); or lastly the amount of entropy that has been produced in our past light cone. All but one of these definitions give a number of gates of order $\Lambda^{-1} \sim 10^{122}$ for our Universe in the present era. (The final definition gives a somewhat lower answer [24] if event horizons are not included.) Thus, for the purposes of this paper, we will take the available resources to be

$$R_{\max} \sim \Lambda^{-1} \tag{3}$$

quantum gates. (Whereas this estimate takes an elementary quantum gate to be the notion of computational step relevant to our Universe, other more speculative possibilities have been considered elsewhere [25–29]).

We note that making the computational censorship hypothesis precise is a difficult problem that we do not claim to have solved. The central difficulty is that our Universe provides us with the solution to one *instance* of a hard problem, whereas computational complexity is defined only for asymptotic families of instances. For any instance of a problem there always exists an efficient algorithm which has the solution to that instance hardwired in.[3] In an intuitive sense, it is clear that the existence of such algorithms is not of interest in determining the difficulty of the instance. Instead we take the complexity of the instance to be the number of steps required by the most efficient general-purpose algorithm that solves it. The distinction between general-purpose algorithms and ones with answers hard-wired seems difficult to formalize, but is typically easy to make in practice.

In the remainder of this paper we will describe various general-purpose number partitioning algorithms that set upper bounds on the complexity of number partitioning problems. Different algorithms provide the best upper bound in different parameter regimes. In all regimes we find that the complexity of the cosmological constant problem within the ADK model is well within the computational capacity of the observable Universe and therefore, contrary to initial appearances based on brute force search, it does not pose a challenge to the computational censorship hypothesis. In some regimes the speedup over brute search achieved by more sophisticated algorithms is quite dramatic; for instances in which the ADK model has 10^9 fields we are able to find a cosmological constant of order 10^{-120} in a few hours on a single processor.

In [30] a model was recently proposed involving a large number of axions, which has the feature that solutions with small cosmological constant are relatively easy to compute. The model of [30] thus provides a way to avoid the computational complexity problems associated with the cosmological constant. Our work shows that, even in models originally cited for their computational difficulty, the complexity problem is less severe than one might naively assume.

Note that the computational Censorship Hypothesis is quite minimal. We require only that some algorithm exists that can solve the problem (e.g., identify a suitable vacuum) in 10^{122} steps or less. We do not require that this algorithm bear any relation to the (largely known) cosmological dynamics that would have produced our Universe. By contrast, recent work of Denef *et al.* explores computational complexity as a possible restriction on the dynamics [31,32]. A related but distinct principle was proposed by Aaronson [19], that NP-complete problems should not be solvable with polynomial resources by any physical means. Recent applications of this and related principles include [29,33,34].

C. An apparent paradox and its resolution

Imposing the computational Censorship Hypothesis leads to an apparent paradox in light of the Denef-Douglas result. To see this, we must quantify the hard problem and show that it requires resources larger than $R_{\max} \sim \Lambda^{-1}$. Indeed, as shown in Sec. II B, the number of elementary computational steps (quantum gates) required to find a solution with $\Lambda \sim 10^{-122}$ by brute force search of the landscape scales as

$$R_{\text{brute}} \sim \Lambda^{-1} (\log_2 \Lambda^{-1})^{3/2}, \tag{4}$$

[2]This method of solving NP-complete problems seems to have been first proposed in [18]; see also [19].

[3]We thank S. Aaronson for stressing this point to us.

which is asymptotically larger than the computational capacity Λ^{-1} in the limit of small Λ. For the particular value of $\Lambda \sim 10^{-122}$, $\Lambda^{-1}(\log_2 \Lambda^{-1})^{3/2}$ exceeds Λ^{-1} by several orders of magnitude.

If the complexity of brute force search were the correct measure of the complexity of the number partitioning problem, then by measuring which vacuum we are in (which is in principle possible, as argued above) we would obtain the solution to an instance of a computational problem which could not be solved within our observable Universe, in violation of the computational censorship hypothesis. Furthermore, this violation does not necessarily require any measurements beyond present-day capabilities. The decision version of the number partitioning problem, of determining whether a solution with residue smaller than a given threshold exists, is already NP-hard, even without demanding that the explicit solution be produced. Thus, if we knew the specifics of the problem instance $(E_0^{(1)}, E_1^{(1)}, \ldots, E_0^{(n)}, E_1^{(n)})$, then the astronomical observations that have already been made, indicating that $\Lambda \simeq 10^{-122}$ already tells us that a residue of that magnitude exists among the solutions to this instance of number partitioning, thereby learning the solution to a large instance of a NP-hard problem.

In the remainder of the paper we will examine how this apparent paradox can be resolved. Our key observation is that modern algorithms can solve the number partitioning problem using far fewer computational steps than are required by brute-force search. The fastest known classical algorithm for general instances of the number partitioning problem runs in $R \sim O(2^{0.291n})$ time [35] and the fastest known quantum algorithm runs in $R \sim O(2^{0.241n})$ time [36]. For $n \lesssim 1300$ these algorithms place the instance of number partitioning arising in the ADK model within the estimated computational capacity of the observable Universe, but far outside the capacity of even the largest supercomputers.

Interestingly, for very large n, the problem becomes solvable with high probability by the Karmarkar-Karp heuristic, which runs in polynomial time,

$$R_{\mathrm{KK}} \sim n \log n, \tag{5}$$

provided that the number of numbers is sufficiently large,

$$n \gtrsim \exp\left[\sqrt{\frac{\log B}{c}}\right], \qquad c \approx 0.7, \tag{6}$$

where B is the typical magnitude of the numbers. In the application to the ADK model,

$$B \sim \Lambda^{-1} \approx 10^{122}. \tag{7}$$

By exploiting the Karmarkar-Karp algorithm, we show in Sec. IV that vacua with $\Lambda \sim 10^{-120}$ can in fact be found in the ADK model in under 3 hours on a standard workstation, provided

$$n \gtrsim 10^9. \tag{8}$$

While the worst case remains NP-hard, Monte Carlo generated average cases can be solved in polynomial time, provided the number of fields is sufficiently large.

In this work we have focused on the ADK model of the landscape, which leads to number partitioning as the underlying computational problem. Karmarkar-Karp is a powerful algorithm against this problem, but it does not generalize to more complex models easily. It will be interesting to investigate the constraints imposed by the computational censorship hypothesis on other toy models, such as the lattice model of BP, which is not amenable to a Karmarkar-Karp-style algorithm. Eventually one would hope to consider a concrete landscape arising from a complete theory, which would dictate both the structure of the partitioning problem and the statistical distribution of the input. For example, the full string landscape [6,37], when its structure becomes better understood, should provide data analogous to the concrete distribution of charges in the BP model.

Our results show that landscape models remain a viable approach to the cosmological constant problem even if the computational censorship hypothesis is adopted. But for now, at least, we cannot confront the hypothesis specifically with the landscape of string theory, for three main reasons. First, the ADK model is purely a toy model; we know of no evidence that it arises from string theory. Second, the string landscape is understood only in a few corners of the theory, where small parameters are available and statistical estimates are arguably under control. In particular, the oft-quoted number 10^{500} of vacua is likely an underestimate [38], and we do not know of a reliable upper bound. Third, even if we did know the structure of the landscape, and supposing that we knew of no general purpose algorithm that satisfied the computational censorship hypothesis, this would not imply that no such algorithm exists.

Outline. In Sec. II we relate the ADK model to number partitioning and estimate the brute force cost of finding a small value of Λ. In Sec. III we review the Karmarkar-Karp and other fast algorithms and discuss their range of applicability. In Sec. IV we report an empirical test of the Karmarkar-Karp algorithm. We demonstrate that it can find a value of Λ consistent with observation in randomly generated instances of an ADK model with nearly 10^9 fields [and so by Eq. (5), in a few hours on a desktop computer]. We find that sieves are less efficient but still suffice to demonstrate consistency with the computational censorship hypothesis.

II. COMPLEXITY OF THE ADK MODEL

In this section, we show that the problem of finding a small cosmological constant Λ in the ADK model can be reduced to the standard number partitioning problem. We then demonstrate that the cost of a brute force search exceeds Λ^{-1} by a factor $(\log_2 \Lambda^{-1})^{3/2}$. Therefore a brute

force search is incompatible with the computational censorship hypothesis.

A. Reduction to number partitioning

The number partitioning problem is, given a list of positive integers $\delta_1, \ldots, \delta_n$, to find

$$\sum_{j=1}^{n} s_j \delta_j = 0, \tag{9}$$

where $s_j \in \{+1, -1\}$. The number partitioning problem is NP-complete[4] and in fact was a member of the list of 21 problems shown to be NP-complete in the 1972 paper of Karp [40], which together with Cook's 1971 paper [41] is credited with founding the theory of NP-completeness.

The problem of finding vacua in the ADK model with cosmological constant 10^{-122} differs superficially from the number partitioning problem in its standard form, but can easily be converted. To do so, first note that we can choose our labels so that for each j, $E_1^{(j)} \geq E_0^{(j)}$. Then, for each $j = 1, \ldots n$ let

$$\delta_j = (E_1^{(j)} - E_0^{(j)})/2 \tag{10}$$

$$\mu_j = (E_1^{(j)} + E_0^{(j)})/2. \tag{11}$$

In this notation, (2) becomes

$$\Lambda = \delta_0 + \sum_{j=1}^{n} s_j \delta_j, \tag{12}$$

where

$$\delta_0 = \sum_{j=1}^{n} \mu_j. \tag{13}$$

It is clear that finding a solution to (12) is very closely related to the number partitioning problem. There are three technical differences. First, the numbers involved are reals rather than integers. This is inconsequential, as reals can be scaled up and rounded to integers, with the scale factor determined by the needed level of precision. Henceforth, we will refer to both the problem of obtaining residue Λ starting with real inputs of order 1 and the problem of obtaining residue 1 starting with integers of order Λ^{-1} as number partitioning, as will be clear from context.

[4]Technically, NP is a class of decision problems. The NP-complete version of the partitioning problem is to decide whether a solution to (2) exists. However, by standard arguments [39], the decision and search versions of the problem are essentially equivalent; the complexity of finding a solution exceeds the complexity of deciding whether one exists by at most a factor of n.

A second difference is that in many works on integer partitioning, one wishes to find a partition in which the residue is zero, rather than merely small. Third, in the problem arising from the ADK model, there is no variable $s_0 \in \{-1, +1\}$ multiplying δ_0. Nevertheless, algorithms that were designed for solving the standard number partitioning problem can be easily adapted to this slight variant of the problem, as we now illustrate.

B. Cost of brute force search

Consider the number partitioning problem on real numbers, where problem instances are generated by drawing n numbers independently at random from the uniform distribution on [0, 1]. In [13] it was proven that the median optimal residue is $\Theta(\sqrt{n}2^{-n})$. (The big-Θ notation indicates that the asymptotic scaling as $n \to \infty$ is $\sqrt{n}2^{-n}$ up to constant factors.) Thus, for a solution with residue Λ to exist, one needs $\sqrt{n}2^{-n} \lesssim \Lambda$. One can show that asymptotically, this means the minimum viable value of n scales as

$$n \sim \log_2 \Lambda^{-1} + \frac{1}{2} \log_2 \log_2 \Lambda^{-1}. \tag{14}$$

To find a residue of size Λ one needs to perform all arithmetic with at least

$$b \sim \log_2 \Lambda^{-1} \tag{15}$$

bits of precision.

A naive method for brute force search would be to increment through all 2^n possible choices of sign $s_1, \ldots, s_n \in \{+1, -1\}$ and for each one, compute the corresponding sum, and compare it against the threshold for sufficient smallness (e.g., 10^{-122}). Such an algorithm would perform $n2^n$ addition (or subtraction) operations, each on b bits. Addition or subtraction of a pair of b-bit numbers can be done by a quantum circuit of $O(b)$ elementary gates [42–47]. Thus the total complexity of this algorithm is $O(nb2^n)$.

However, there is a somewhat more efficient algorithm that still arguably qualifies as brute force search. Rather than summing up the residue from scratch with each new choice of signs, one could use the residue from the previous calculation and add or subtract $2\delta_j$ for each j in which the sign has changed. For any n there always exists an ordering of the 2^n bit strings of length n such that each bit string is obtained from the previous one by only flipping a single bit. These orderings are called Gray codes, and they can furthermore be generated by efficient classical algorithms [48]. By ordering the choices of sign according to a Gray code one thus has to do n additions on the first step, and only one addition or subtraction on each of the subsequent $2^n - 1$ steps. This brings the total complexity of the algorithm down to $O(b2^n)$ elementary quantum gates.

By (14) and (15) this yields a total complexity of order $\Lambda^{-1}(\log_2 \Lambda^{-1})^{3/2}$ quantum gates.[5]

III. ALGORITHMS FOR NUMBER PARTITIONING

In this section, we discuss efficient algorithms for the number partitioning problem.

The number partitioning problem is NP-complete. Assuming $P \neq NP$ this implies that no polynomial-time classical algorithm can solve all instances of number partitioning in time scaling polynomially in n. However, this does not forbid the existence of parameter regimes in which classical algorithms can solve the problem in polynomial time. In fact, for many NP-complete problems, including the canonical example of 3-SAT, randomly generated instances are efficiently solvable generically; exponentially hard instances require fine-tuning [49].

Random instances of number partitioning have been well studied using methods of statistical mechanics. The standard ensemble of instances most typically studied is to set some magnitude parameter B and then choose n integers $\delta_1, \ldots, \delta_n$ independently uniformly at random from the range $\{1, 2, \ldots, B\}$. If $\sum_{j=1}^n \delta_j \equiv 1 \bmod 2$ then any sum of the form $\sum_{j=1}^n \pm\delta_j$ will be odd, and it is impossible for a solution to (9) to exist. Thus, it is conventional to define a perfect partition as a solution to (9) in the case that $\sum_{j=1}^n \delta_j$ is even, and as a solution to $\sum_{j=1}^n s_j\delta_j = 1$ in the case that $\sum_{j=1}^n \delta_j$ is odd. Whether a perfect partition exists for an instance of number partitioning sampled from the standard ensemble depends on the relationship between n and B. If n is too small relative to B then the system is overconstrained and is likely to have no perfect partitions, whereas if n is sufficiently large relative to B then the system is underconstrained and is likely to have many perfect partitions. More precisely, as shown in [50], in the limit of large n, randomly generated number partitioning problems will have no perfect partitions for $B > 2^{n+O(\log n)}$ and will have exponentially many partitions for $B < 2^{n+O(\log n)}$. As is the case for many NP-complete problems, the number partitioning problem becomes easier for instances sufficiently far from the phase transition.

For example, the Karmarkar-Karp algorithm solves number partitioning in time $O(n \log n)$ for $B < n^{c \log n}$, which is to say when $n > \exp[\sqrt{\frac{\log B}{c}}]$ for some constant c. It was proven rigorously in [51] that $c = \frac{1}{2 \log 2} = 0.721\ldots$ suffices. In Sec. IV we empirically achieve success with $c = 0.662$, which is in rough agreement with the empirical testing in [52]. Nonetheless, the statistical mechanics arguments in [52] suggest that $c = 0.721$ is the true asymptotic value as $n \to \infty$.

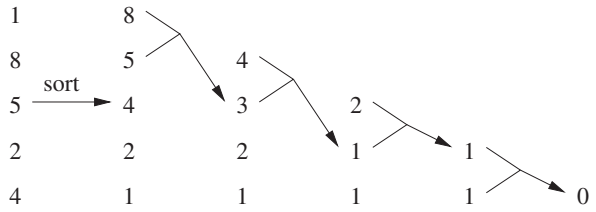

FIG. 1. An example of the Karmarkar-Karp algorithm. At the first step the numbers are sorted. At each subsequent step, the largest two numbers are replaced by their difference, which is then inserted into the appropriate location in the list so that it remains sorted. The sequence of moves in the example shown finds the solution $1 - (2 - (4 - (8 - 5))) = 0$.

A. The Karmarkar-Karp algorithm

The Karmarkar-Karp algorithm is based on the intuition that the largest numbers should be given opposite signs in order to achieve cancellation. The Karmarkar-Karp strategy is to commit to giving the largest two numbers opposite signs without specifying which should be positive and which should be negative. This reduces the problem to a new instance of integer partitioning with one fewer number: the largest two numbers have been replaced by their difference. This is then treated in the same manner, until only one number is left, which is the final residue $\sum_{i=1}^n s_i\delta_i$. An example is given in Fig. 1.

The initial sorting step has complexity $O(n \log n)$ by standard algorithms. Inserting a number into the correct location in an ordered list can be achieved with complexity $O(\log n)$ using a standard data structure called a heap [53]. There are exactly $n - 1$ differencing-and-insertion steps needed to arrive at a final residue. Thus the total complexity of the algorithm is $O(n \log n)$.

The Karmarkar-Karp algorithm is heuristic in the sense that for some problem instances for which a perfect partition exists, the Karmarkar-Karp algorithm will fail to find it. On the other hand, as mentioned earlier, for random instances of integer partitioning with $B < n^{0.721 \log n}$, the Karmarkar-Karp algorithm will succeed with probability going to 1 as $n \to \infty$ [51]. Korf [54] has introduced an extension of the Karmarkar-Karp algorithm, which initially proceeds identically to the Karmarkar-Karp algorithm and terminates if this yields a perfect partition. However, if it fails to find a perfect partition it continues searching by backtracking and trying assignments in which the largest two numbers are given the same sign. The details of Korf's algorithm are such that it is guaranteed to find a perfect partition provided one exists. For $B < n^{c \log n}$ Korf's algorithm matches the performance of the Karmarkar-Karp algorithm, but for $B \gg n^{c \log n}$ it may have an exponentially long runtime.

Other heuristic algorithms derived from Karmarkar-Karp were studied in [55], where it was empirically found that, in

[5]In a more realistic model, the contributions to the vacuum energy from various fields have to be recomputed in every vacuum, adding further overhead to the calculation. Since the matter sector can be more complex for a small cosmological constant, one expects this overhead to grow at least weakly with Λ^{-1}.

the regime where Karmarkar-Karp finds a residue much larger than the optimal residue, modest improvements in residue size can be obtained by exhaustively or stochastically searching for solutions "near" the Karmarkar-Karp solution, if the notion of nearness is carefully chosen. However, other than near the Karmarkar-Karp solution, the optimization landscape in number partitioning problems was found to be hard to distinguish from random, based on any of the neighborhood notions that were investigated. Thus there appears to be little structure in the problem for general-purpose optimization heuristics such as simulated annealing or genetic algorithms to exploit. This is corroborated by the relatively modest performance improvements obtained by such heuristics on number partitioning in other studies [56–58].

In analyzing the performance of the Karmarkar-Karp algorithm it is standard to consider the ensemble of instances where the $\delta_1, \ldots, \delta_n$ are independent, identically distributed random variables, typically sampled from a uniform distribution on some range 0 to B. The instances of number partitioning arising in the context of the ADK model may slightly differ from this. In particular, from Eqs. (10) through (13), one sees that if $E_1, \ldots, E_n$ are each of order B, then $\delta_1, \ldots, \delta_n$ will be of order B, but δ_0 will generically be of order $\sqrt{n}B$. It is easy to see that this makes only a small difference to the performance of the Karmarkar-Karp algorithm. The first $\sim\sqrt{n}$ differencing steps will all be used to difference from δ_0. After that, one is left with a standard instance of integer partitioning in which all the numbers are of similar magnitude, and the Karmarkar-Karp algorithm performs as it would on the standard ensemble. Thus, whereas for the standard ensemble, one would have required a minimum of $n_{\min}^{\text{std}} \simeq \exp[\sqrt{\frac{\log \Lambda^{-1}}{c}}]$, the minimum number of fields in the ADK case may be slightly larger: $n_{\min}^{\text{ADK}} \simeq n_{\min}^{\text{std}} + \sqrt{n_{\min}^{\text{std}}}$.

In Sec. IV we give the results of some computer experiments on the performance of the Karmarkar-Karp algorithm, confirming the predictions of the statistical analyses referenced above and giving a quantitative sense of the practical performance of the algorithm. For simplicity, and to facilitate comparison with the existing literature, the experiments in Sec. IV are performed using a standard ensemble of instances of number partitioning.

B. Dynamic programming

The computational difficulty of the number partitioning problem depends on the number of numbers n, and their magnitudes. In the regime where the $B = \max_j \delta_j$ is only polynomially large, i.e., the number of bits needed to represent the numbers scales only as some power of $\log n$, the number partitioning problem can be solved in polynomial time on classical computers using a standard technique called dynamic programming. Specifically, as is described nicely in Sec. IV.2 of [59], dynamic programming solves the number partitioning problem in time $\tilde{O}(nD)$, where $D = \sum_{j=1}^{n} \delta_j$. Problems such as number partitioning that can be solved in polynomial time when all the input numbers are restricted to polynomial magnitude (rather than allowing them to be polynomially many bits long) are said to be pseudopolynomial [60].

C. Adapting algorithms for subset sum

Number partitioning, subset sum, and knapsack problems are all variants of essentially the same problem. Algorithms for one are often applicable, with minor modification, to the others. For example, a straightforward meet-in-the-middle tree search [61] applies to all these problems and succeeds in finding the optimal residue in time $\approx 2^{0.5n}$. At present, the *asymptotically* best upper bound on the classical complexity of finding the optimal solution to number partitioning problems is given by the algorithm of [35], which is guaranteed to succeed in time $O(2^{0.291n})$. The asymptotically best upper bound on the quantum complexity of this problem is given by the quantum algorithm of [36], which is guaranteed to find the optimum using a number of elementary steps (quantum gates) at most $O(2^{0.241n})$. (This quantum algorithm is based on quantum walks. An adiabatic quantum algorithm for this problem has also been analyzed, but its

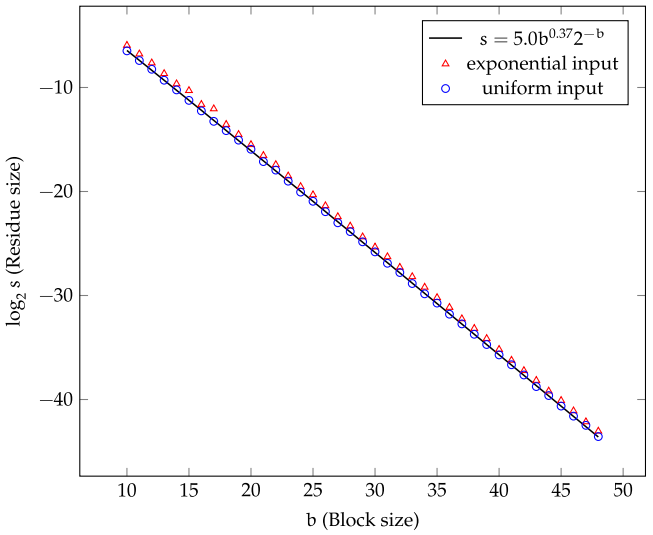

FIG. 2. Expected relative optimal residue size versus input size for number partitioning problems on a block of b random numbers with the specified distribution. For each block size b in this given range, mean size for 1000 experiments is shown. Each experiment generated high precision floating point input data with mean one and a complete NPP solver produced the optimal residue. Assuming the distribution of optimal residues is exponential, the maximum likelihood estimator of the mean is biased; hence the least square error estimator was used to find the mean in each case. The model parameters with residue size $s = 5.0b^{0.37}2^{-b}$ were generated by linear regression on the data with uniformly distributed inputs.

runtime is not known. Numerical calculations in [62] suggest a runtime scaling as $2^{0.8n}$. The adiabatic algorithm may also be limited in its capacity to accommodate large B).

As discussed in Sec. II B, the minimum value of n such that the number partitioning problem is likely to have a solution of order Λ is asymptotically $\log_2 \Lambda^{-1} + \frac{1}{2}\log_2 \log_2 \Lambda^{-1}$. The algorithm of [35] could solve a problem of this size with runtime of order $(\Lambda^{-1})^{0.291}(\log_2 \Lambda^{-1})^{0.146}$.

D. Adapting lattice sieves

Here we explore a very simple sieve mechanism for solving the number partitioning problem inspired by "lattice sieves" [63]. The Karmarkar-Karp algorithm can be viewed as a form of the Gauss sieve [64] for a one-dimensional lattice. Curiously, while more sophisticated lattice sieves easily outperform the Gauss sieve on high-dimensional lattices [65–69], here we find that this is seemingly not the case for the number partitioning problem. The simple sieve we present here is similar in spirit to the "tuple sieve" of [69], but cannot match the performance of the Karmarkar-Karp algorithm as we will show. Nonetheless, the key advantage of this style of sieve is that it is not restricted to the number partitioning problem and so could be easily adapted to other models of the landscape.

In general, a sieve consists of several stages. For us, the input to a stage is a collection of numbers; these are partitioned into small blocks of size b and on each of these blocks the number partition problem is solved for the optimal residue. This collection of residues is the output of the sieve stage, which then becomes the input for the next stage. There are a number of algorithms to solve for the optimal residue, some of which are illustrated in the previous sections. All of these take work $2^{\alpha b+o(b)}$. As long as the distribution of the input data is sufficiently well behaved, the optimal residues will be exponentially distributed with expected size $2^{-b+o(b)}$, asymptotically $O(\sqrt{b}2^{-b})$ [13–15]. In Fig. 2, we validate this scaling for small b but recover a smaller power in the polynomial factor in this formula. In Fig. 3, we also validate that the distribution of the residues is well modeled as exponential with the parameter λ estimated from the data.

If our input is n fields producing mean energy differences $\delta_j \approx 1$, the first sieve stage involves solving n/b_1 number partition problems, each of size b_1. The work for this stage is $\approx \frac{n}{b_1}2^{\alpha b_1}$ and the output is $\frac{n}{b_1}$ residues exponentially distributed with mean size $\approx 2^{-b_1}$. The second sieve stage partitions these into blocks of size b_2 and solves the number partition problem on each to produce $\frac{n}{b_1 b_2}$ residues of size $\approx 2^{-(b_1+b_2)}$, and so on.

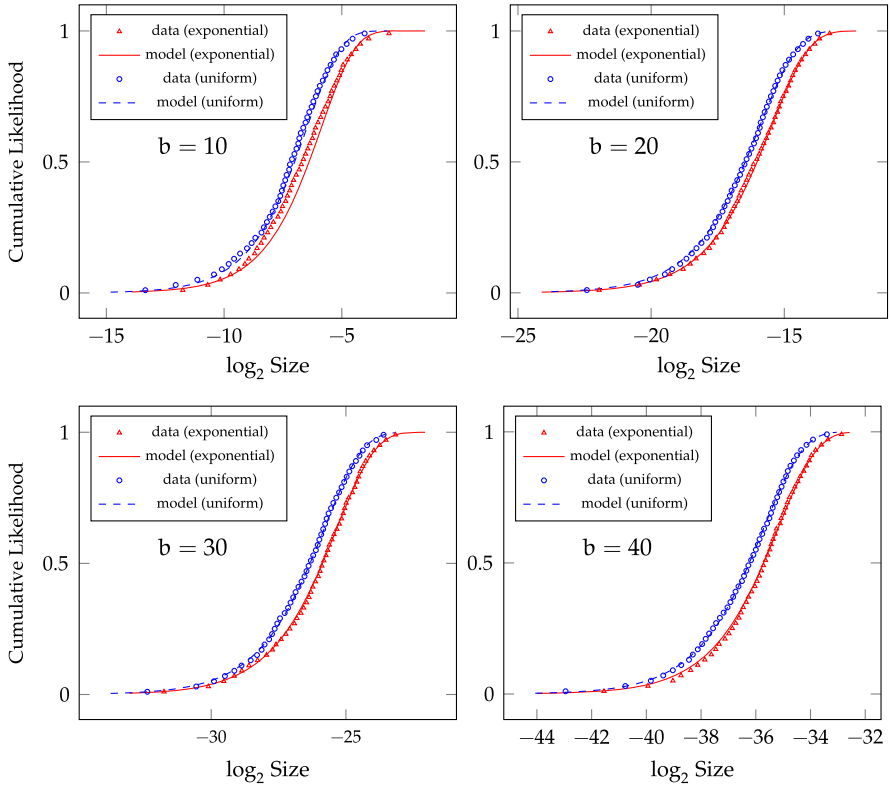

FIG. 3. Plots of cumulative likelihood of observing the optimal residue versus (log) size of the optimal residue. The model is the cumulative distribution function of the exponential distribution where the single parameter λ is computed from the data using the least squares estimator. For block sizes $b = 10, 20, 30, 40$, each plot was generated from high precision floating point input data (uniformly or exponentially distributed) with mean one and a complete solver produced the optimal residue.

TABLE I. Example sieves for $k=2,\ldots,8$ stages with overall expected residue $2^{-t}\approx 2^{-400}$. The sizes of the blocks $(n_1,n_2,\ldots)$ are selected so the overall work in each stage is approximately equal. In this range as the number of layers increases, the required number of input fields n increases, and the overall work of the sieve 2^w decreases. However, at smaller block sizes (for instance $b_1=16$ for $k=8$), variations in the size of the resulting residues are large and so the work estimates given are less accurate.

k	n	t	w	$(n_1,n_2,\ldots)$
2	4.22×10^4	400.0	107.62	(198, 213)
3	2.65×10^6	400.8	78.32	(124, 139, 154)
4	1.19×10^8	400.8	65.07	(85, 98, 113, 126)
5	3.96×10^9	400.0	58.14	(59, 72, 85, 98, 112)
6	1.03×10^{11}	400.3	54.53	(41, 53, 65, 77, 91, 104)
7	1.97×10^{12}	400.8	52.70	(27, 38, 49, 61, 74, 87, 100)
8	2.54×10^{13}	400.5	51.88	(16, 26, 36, 48, 59, 72, 85, 98)

The goal is that after k sieve stages we produce a single residue of expected length $2^{-t}\approx 2^{-(b_1+\cdots+b_k)}$. The optimal work is given when we follow an "equipartition principle" and balance the amount of work done on each sieve stage. For example, the first sieve stage involves solving many more number partition problems than the second stage, and so we should choose $b_2 > b_1$ so as to balance the amount of work done during the first stage with that done in the second. Specifically, in stage $j \le k$ of the sieve, we solve $n/(b_1\cdots b_j)$ number partition problems with an overall work of $n/(b_1\cdots b_j)2^{\alpha b_j}$, which we balance with the work in stage $j-1$:

$$\frac{n}{b_1\cdots b_j}2^{\alpha b_j}\approx\frac{n}{b_1\cdots b_{j-1}}2^{\alpha b_{j-1}}. \tag{16}$$

Therefore we select b_j implicitly by solving

$$b_j-\frac{1}{\alpha}\log_2(b_j)\approx b_{j-1}. \tag{17}$$

The overall work of the sieve is then $\sim\frac{kn}{b_1}2^{\alpha b_1}$. Examples of sieves for $k=2,\ldots,8$ stages, $\alpha=0.5$, all targeting residues of size $\approx 2^{-400}$, are given in Table I.

This table indicates that only the sieves with $k=2,3,4$ can outperform Karmarkar-Karp in terms of the number of fields, which requires $n\approx 8\times10^8$ to produce residues of size $\approx 2^{-400}$. At this size Karmarkar-Karp takes work roughly 2^{35}, well below that of any of these sieves. To outperform Karmarkar-Karp with this style of sieve, the algorithm that solves number partitioning on the blocks would need to have $\alpha\lesssim 0.22$, and even then lower-order terms not counted in the asymptotic expression would likely dominate the work.

IV. COMPUTER EXPERIMENTS

In this section, we apply fast algorithms to the problem of finding a small cosmological constant in an ADK landscape. We show that they allow the computational censorship hypothesis to be satisfied.

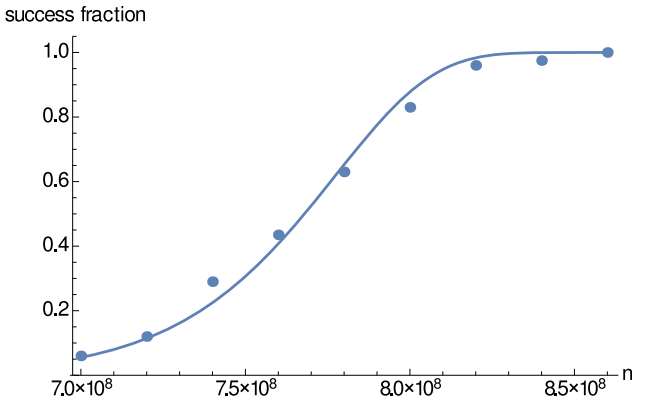

FIG. 4. At each value of n, 200 instances of number partitioning are generated with each of the n numbers independently sampled uniformly from $\{0,1,2,\ldots,2^{430}-1\}$. The fraction of instances in which the Karmarkar-Karp algorithm found a residue smaller than 2^{30} is shown for each n. The theoretically predicted success probability of $1-\exp[-\frac{e^{c\log(n)^2}}{2^{400}}]$ is also shown, with $c=0.6615$ determined by fitting to the data. The asymptotic value of c as $n\to\infty$ is predicted to be $1/\sqrt{2}\simeq 0.7071$.

A. Karmarkar-Karp

To empirically test the Karmarkar-Karp algorithm in a regime relevant to the cosmological constant problem, we generated random instances of the number partitioning problem, at various values of n, in which each of the n numbers are independently sampled uniformly from $\{0,1,2,\ldots,2^{430}-1\}$. In Fig. 4, we plot the fraction of instances on which the Karmarkar-Karp algorithm was successful with n numbers, where we defined success as achieving residue less than 2^{30}. In the context of finding a small cosmological constant within the ADK model, one starts with real numbers of order 1 and seeks to find a residue of order 10^{-122}. Here we have scaled up the numbers by a factor of 2^{430} and represented them as integers. This use of fixed-point arithmetic is strictly for

TABLE II. Predicted parameters of a four stage sieve for the number partition problem. Upon input of 1.2×10^6 numbers uniformly distributed on [0, 1], four stages of sieving outputs a single number of expected magnitude $2^{-121.3}$. The sieve is balanced so that the amount of computation spent during each stage is roughly equal.

Stage	b	Inputs distribution	Number of NPPs	Work	$\mathbb{E}[s]$
1	20	1200000 uniform	60000	$2^{25.9}$	$2^{-16.1}$
2	30	60000 exponential	2000	$2^{26.0}$	$2^{-41.3}$
3	40	2000 exponential	50	$2^{25.6}$	$2^{-76.4}$
4	50	50 exponential	1	$2^{25.0}$	$2^{-121.3}$

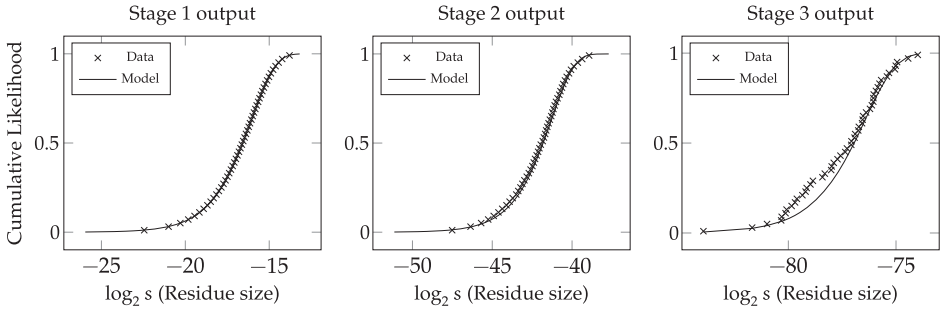

FIG. 5. Plots of cumulative likelihood of observing the optimal residue versus (log) size of the optimal residue for a four-stage sieve. Input to stage 1 was $n = 1.20 \times 10^6$ mean one uniformly distributed numbers. Stage 1 combined $b_1 = 20$ numbers in each number partitioning problem to produce 60000 optimal residues, forming the input to stage 2. Stage 2 combined $b_2 = 30$ numbers in each problem to produce 2000 optimal residues. Stage 3 combined $b_3 = 40$ numbers to produce 50 optimal residues. Finally stage 4 combined these $b_4 = 50$ numbers to produce an overall residue of 6.54×10^{-38}. This final residue was slightly smaller than the predicted $2^{-121.3}$. The sieve completed in 152 seconds on a standard desktop computer.

computational convenience. Our definition of success corresponds to achieving a residue that is smaller than the magnitude of the initial numbers by a factor of $2^{400} \simeq 10^{120}$ and thus corresponds to finding a cosmological constant close to that observed for our Universe.[6] The extra 30 bits of precision are to ensure that "numerical noise" should be small.

By the analysis of [52], if the Karmarkar-Karp algorithm is applied to real numbers uniformly distributed on [0, 1], the size of the final residue should be exponentially distributed. That is, the probability that the residue lies between y and $y + dy$ should be $\lambda e^{-\lambda y} dy$, where

$$\lambda = e^{-c \log^2 n} \tag{18}$$

and c is asymptotically equal to $1/\sqrt{2}$ as $n \to \infty$. Empirical studies at finite n consistently observe values of c smaller than $1/\sqrt{2}$ [52]. By defining success to be a reduction factor of $\epsilon = 2^{-400}$, we should obtain the success probability

$$P = \int_0^{\epsilon} \lambda e^{-\lambda y} dy \tag{19}$$

$$= 1 - \exp\left[-e^{-c \log^2 n} \epsilon\right]. \tag{20}$$

[6]A more precise match to our Universe would be to seek a factor of 2^{406}, but this was not convenient to work with because it put the memory requirements of the algorithm just slightly beyond the available 128G of RAM on most of our computers. Achieving a factor 2^{406} requires $n \simeq 8.7 \times 10^8$ and correspondingly an increase in time and memory cost of less than 20%.

As one can see from Fig. 4, the observed success fraction from our trials of the Karmarkar-Karp algorithm on random instances agrees well with this prediction if we take $c = 0.6615$.

B. Sieves

The predicted work of a sieve to produce a residue of length 2^{-400} is not so large that the Universe would be unable to compute it, but it is large enough to require significant effort with current hardware. As a simple proof of concept, we will tackle a scaled down version with four sieve stages of block sizes $(b_1, b_2, b_3, b_4) = (20, 30, 40, 50)$, and use a simple meet-in-the-middle algorithm ($\alpha = 0.5$) to solve the number partitioning problem [61]. The profile of the experiment can be found in Table II, which predicts an expected size of the final residue output at sieve stage 4 to be $\mathbb{E}[s] = 2^{-121.3}$.

The result of the experiment is captured in Fig. 5.

ACKNOWLEDGMENTS

We would like to thank Scott Aaronson, Adam Bouland, and Liam McAllister for discussions. N. B. is supported in part by the DuBridge Fellowship of the Walter Burke Institute for Theoretical Physics. R. B. is supported in part by the Berkeley Center for Theoretical Physics, by the National Science Foundation (Grants No. PHY-1521446 and No. PHY-1316783), by FQXi, and by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. S. J. and B. L. thank U. Maryland for use of the *Deepthought2* high performance computing cluster. Parts of this manuscript are a contribution of NIST, an agency of the U.S. government, and are not subject to U.S. copyright.

[1] S. Perlmutter *et al.*, Measurements of Omega and Lambda from 42 high-redshift supernovae, Astrophys. J. **517**, 565 (1999).

[2] A. G. Riess *et al.*, Observational evidence from supernovae for an accelerating universe and a cosmological constant, Astron. J. **116**, 1009 (1998).

[3] P. A. R. Ade *et al.*, Planck 2015 results. XIII. Cosmological parameters, Astron. Astrophys. **594**, A13 (2016).

[4] J. Polchinski, The cosmological constant and the string landscape, arXiv:hep-th/0603249.

[5] R. Bousso, TASI lectures on the cosmological constant, Gen. Relativ. Gravit. **40**, 607 (2008).

[6] R. Bousso and J. Polchinski, Quantization of four-form fluxes and dynamical neutralization of the cosmological constant, J. High Energy Phys. 06 (2000) 006.

[7] G. W. Gibbons and S. W. Hawking, Cosmological event horizons, thermodynamics, and particle creation, Phys. Rev. D **15**, 2738 (1977).

[8] R. Bousso, A covariant entropy conjecture, J. High Energy Phys. 07 (1999) 004.

[9] S. Weinberg, Anthropic Bound on the Cosmological Constant, Phys. Rev. Lett. **59**, 2607 (1987).

[10] R. Bousso, B. Freivogel, S. Leichenauer, and V. Rosenhaus, A Geometric Solution to the Coincidence Problem, and the Size of the Landscape as the Origin of Hierarchy, Phys. Rev. Lett. **106**, 101301 (2011).

[11] F. Denef and M. R. Douglas, Computational complexity of the landscape I, Ann. Phys. (Amsterdam) **322**, 1096 (2007).

[12] N. Arkani-Hamed, S. Dimopoulos, and S. Kachru, Predictive landscapes and new physics at a TeV, arXiv:hep-th/0501082.

[13] N. Karmarkar, R. M. Karp, G. S. Leuker, and A. M. Odlyzko, Probabilistic analysis of optimum partitioning, J. Appl. Probab. **23**, 626 (1986).

[14] S. Mertens, Random Costs in Combinatorial Optimization, Phys. Rev. Lett. **84**, 1347 (2000).

[15] H. Bauke, S. Franz, and S. Mertens, Number partitioning as a random energy model, J. Stat. Mech. 2004, P04003 (2004).

[16] R. Bousso, Complementarity in the multiverse, Phys. Rev. D **79**, 123524 (2009).

[17] R. Bousso and L. Susskind, The multiverse interpretation of quantum mechanics, Phys. Rev. D **85**, 045007 (2012).

[18] H. Moravec, *Mind Children* (Harvard University Press, Cambridge, MA, 1990).

[19] S. Aaronson, NP-complete problems and physical reality, ACM SIGACT News **36**, 30 (2005).

[20] A. R. Brown, D. A. Roberts, L. Susskind, B. Swingle, and Y. Zhao, Complexity, action, and black holes, Phys. Rev. D **93**, 086006 (2016).

[21] S. Lloyd, Computational Capacity of the Universe, Phys. Rev. Lett. **88**, 237901 (2002).

[22] R. Bousso, Holography in general space-times, J. High Energy Phys. 06 (1999) 028.

[23] R. Bousso, Positive vacuum energy and the N-bound, J. High Energy Phys. 11 (2000) 038.

[24] R. Bousso, R. Harnik, G. D. Kribs, and G. Perez, Predicting the cosmological constant from the causal entropic principle, Phys. Rev. D **76**, 043513 (2007).

[25] S. Aaronson and J. Watrous, Closed timelike curves make quantum and classical computing equivalent, Proc. R. Soc. A **465**, 631 (2009).

[26] C. H. Bennett, D. Leung, G. Smith, and J. A. Smolin, Can Closed Timelike Curves or Nonlinear Quantum Mechanics Improve Quantum State Discrimination or Help Solve Hard Problems?, Phys. Rev. Lett. **103**, 170502 (2009).

[27] T. A. Brun and M. M. Wilde, Simulations of closed timelike curves, Found. Phys. **47**, 375 (2017).

[28] D. S. Abrams and S. Lloyd, Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and #P Problems, Phys. Rev. Lett. **81**, 3992 (1998).

[29] N. Bao, A. Bouland, and S. P. Jordan, Grover Search and the No-Signaling Principle, Phys. Rev. Lett. **117**, 120501 (2016).

[30] T. C. Bachlechner, K. Eckerle, O. Janssen, and M. Kleban, Axions of evil, arXiv:1703.00453.

[31] F. Denef, M. R. Douglas, B. Greene, and C. Zukowski, Computational complexity of cosmology in string theory, Talk given by *M. R. Douglas at JHS 75, Caltech, November 2016*, slides and video available at https://burkeinstitute.caltech.edu/workshops/JHS75; Talk given by *M. R. Douglas at "New Horizons in Inflationary Cosmology," Stanford SITP, March 2017*, slides and video available at https://sitp.stanford.edu/conferences/new-horizons-inflationary-cosmology.

[32] F. Denef, M. R. Douglas, B. Greene, and C. Zukowski, Computational complexity of the landscape II—Cosmological considerations, arXiv:1706.06430.

[33] D. Harlow and P. Hayden, Quantum computation vs. firewalls, J. High Energy Phys. 06 (2013) 085.

[34] N. Bao, A. Bouland, A. Chatwin-Davies, J. Pollack, and H. Yuen, Rescuing complementarity with little drama, J. High Energy Phys. 12 (2016) 026.

[35] A. Becker, J.-S. Coron, and A. Joux, Improved generic algorithms for hard knapsacks, in *Advances in Cryptology–EUROCRYPT 2011*, LNCS Vol. 6632 (Springer, New York, 2011), pp. 364–385.

[36] D. J. Bernstein, S. Jeffery, T. Lange, and A. Meurer, Quantum algorithms for the subset-sum problem, in *PQCrypto 2013: Post-Quantum Cryptography 2013*, LNCS Vol. 7932 (Springer, New York, 2013), pp. 16–33.

[37] S. Kachru, R. Kallosh, A. Linde, and S. P. Trivedi, De Sitter vacua in string theory, Phys. Rev. D **68**, 046005 (2003).

[38] M. R. Douglas and S. Kachru, Flux compactification, Rev. Mod. Phys. **79**, 733 (2007).

[39] C. Papadimitriou, *Computational Complexity* (Addison-Wesley, Reading, MA, 1994).

[40] R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, edited by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger (Springer US, Boston, MA, 1972), pp. 85–103.

[41] S. Cook, The complexity of theorem proving procedures, in *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC)* (ACM, New York, 1971), pp. 151–158.

[42] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, Phys. Rev. A **54**, 147 (1996).

[43] T. G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033.
[44] T. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore, A logarithmic-depth quantum carry-lookahead adder, Quantum Inf. Comput. **6**, 351 (2006).
[45] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton, A new quantum ripple-carry addition circuit, arXiv:quant-ph/0410184.
[46] Y. Takahashi, S. Tani, and N. Kunihiro, Quantum addition circuits and unbounded fan-out, Quantum Inf. Comput. **10**, 872 (2010).
[47] F. Wang, M. Luo, H. Li, Z. Qu, and X. Wang, Improved quantum ripple-carry addition circuit, Sci. China Inform. Sci. **59**, 042406 (2016).
[48] R. W. Doran, The Gray code, Journal of Universal Computer Science **13**, 1573 (2007).
[49] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Determining computational complexity from characteristic phase transitions, Nature (London) **400**, 133 (1999).
[50] S. Mertens, Phase Transition in the Number Partitioning Problem, Phys. Rev. Lett. **81**, 4281 (1998).
[51] B. Yakir, The differencing algorithm LDM for partitioning: A proof of a conjecture of Karmarkar and Karp, Math. Oper. Res. **21**, 85 (1996).
[52] S. Boettcher and S. Mertens, Analysis of the Karmarkar-Karp differencing algorithm, Eur. Phys. J. B **65**, 131 (2008).
[53] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, MA, 2009).
[54] R. E. Korf, A complete anytime algorithm for number partitioning, Artif. Intell. **106**, 181 (1998).
[55] W. Ruml, J. T. Ngo, J. Marks, and S. M. Schieber, Easily searched encodings for number partitioning, J. Optim. Theory Appl. **89**, 251 (1996).
[56] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevron, Optimization by simulated annealing: An experimental evaluation. Part II, Graph coloring and number partitioning, Oper. Res. **39**, 378 (1991).
[57] M. F. Arguello, T. A. Feo, and O. Goldshmidt, Randomized methods for the number partitioning problem, Computers and Operations Research **23**, 103 (1996).
[58] R. E. Berretta, P. Moscato, and C. Cotta, Enhancing a memetic algorithms' performance using a matching-based recombination algorithm: Results on the number partitioning problem, *Metaheuristics: Computer Decision-Making*, edited by M. G. C. Resende and J. Souza (Kluwer Academic Publishers, Dordrecht, 2004), pp. 65–90.
[59] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman, San Francisco, 1979).
[60] S. Mertens, The easiest hard problem: Number partitioning, in *Computational Complexity and Statistical Physics*, edited by A. Percus, G. Istrate, and C. Moore (Oxford University Press, New York, 2006), Chap. 5.
[61] E. Horowitz and S. Sahni, Computing partitions with applications to the knapsack problem, J. Assoc. Comput. Mach. **21**, 277 (1974).
[62] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, What Is the Computational Value of Finite-Range Tunneling?, Phys. Rev. X **6**, 031015 (2016).
[63] M. Ajtai, R. Kumar, and D. Sivakumar, A sieve algorithm for the shortest lattice vector problem, in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing* (ACM, New York, 2001), pp. 601–610.
[64] D. Micciancio and P. Voulgaris, Faster exponential time algorithms for the shortest vector problem, in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, 2010), pp. 1468–1480.
[65] F. Zhang, Y. Pan, and G. Hu, A three-level sieve algorithm for the shortest vector problem, in *International Conference on Selected Areas in Cryptography* (Springer, New York, 2013), pp. 29–47.
[66] A. Becker, N. Gama, and A. Joux, A sieve algorithm based on overlattices, J. Assoc. Comput. Mach. **17**, 49 (2014).
[67] T. Laarhoven and B. de Weger, Faster sieving for shortest lattice vectors using spherical locality-sensitive hashing, in *International Conference on Cryptology and Information Security in Latin America* (Springer, New York, 2015), pp. 101–118.
[68] A. Becker, N. Gama, and A. Joux, Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search, IACR Cryptology ePrint Archive **2015**, 522 (2015).
[69] S. Bai, T. Laarhoven, and D. Stehlé, Tuple lattice sieving (to be published).