Optimization algorithms and the cosmological constant

Ning Bao Raphael Bousso Stephen Jordan Brad Lackey*

arXiv:1706.08503

1 August 2017

Joint Center for Quantum Information and Computer Science

UMIACS University of Maryland Institute for Advanced Computer Studies

University of Maryland

Outline

Outline

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 $\Lambda \approx 1.5 \times 10^{-123} M_P^4$;[1]
- consistency with well-established cosmological history severely constrains large classes of approaches to this problem.[2]

In a landscape model,

- the universe can form large regions with different values of Λ;
- there are exponentially many ways of constructing a "vacuum;"
- observers necessarily find themselves in a highly atypical region that allows for a larger cosmological horizon.

Consistent with standard cosmological history if neighboring vacua have very different energies.[3]

[1] Perlmutter, et al., *Astrophys. J.*, 1999; Riess, et al., *Astro. J.*, 1998; Ade, et al., *Astron. Astrophys.*, 2016.

[2] Polchinski, hep-th/0603249; Bousso, *Gen. Rel. Grav.*, 2008.

[3] Bousso, Polchinski, *J. High Energy Physics*,, 2000.

Models of the landscape

Two simplified models of the landscape capture essential features:

- Arkani-Hamed-Dimopolous-Kachru (ADK) model[4], and
- Bousso-Polchinski (BP) model[5].

Here we focus on a simplification of the ADK model:

- the cosmological constant is obtained by summing the energy contributions from a large number of fields;
- each field is subject to a double-well potential;
- the two minima of each field to be a random number with mean zero and deviation of of order 1 in Planck units.

Given n such fields where vacuum energies $E_0^{(j)}$ and $E_1^{(j)}$, there are 2^n vacua, specified by $s \in \{0,1\}^n$:

$$\Lambda[s(j)] = \sum_{j=1}^{n} E_{s(j)}^{(j)}.$$

[4] Arkani-Hamed, Dimopolous, Kachru, hep-th/0501082

[5] Bousso, Polchinski, ibid.

Complexity

The ADK model is a variant of the number partitioning problem.

- This class of problems in NP-complete.

What cosmological dynamics solved the “hard” problem?

- The universe is exponentially expanding, creating new regions;
- gravity supplies resources for solving the problem;
- observers necessarily find themselves in the regions where a large problem has been solved.

Or, a local viewpoint trades the multiverse for “many worlds”

- one considers the different decay chains through the landscape;
- a patch decoheres rapidly when a vacuum transition takes place;
- observers find themselves in a branch that produced a vacuum with small Λ.

Computational censorship

Computational Censorship Hypothesis:

- a physical measurements should not access the solution to a problem that could not have been solved by the physical resources in the observable universe.

Possible definition of "resources" include:

- the Einstein-Hilbert-matter action,[6]
- the energy of the universe times its age[7];
- the maximum entropy of the visible universe;[8]
- the amount of entropy produced in our past light-cone.[9]

All given a number of gates $\Lambda^{-1} \approx 10^{122}$ (or slightly lower).

[6] Brown, Roberts, Susskind, Swingle, Zhao, *Phys. Rev. D*, 2016.

[7] Lloyd, *Phys. Rev. Lett.*, 2002

[8] Bousso, *JHEP*, 1999.

[9] Bousso, Harnik, Kribs, Perez, *Phys. Rev. D*, 2007.

Resolution

This leads to an "apparent paradox" in the ADK model.

- Resources available are $\approx \Lambda^{-1}$.
- Brute force search of the landscape scales as $\sim \Lambda^{-1} \left(\log_2 \Lambda^{-1}\right)^{3/2}$.

However this assumes n (number of fields in the ADK model) is such that Λ is an optimal solution to number partitioning.

For very large n, there are polynomial time (in n) heuristics.

1. There is no known way to bound how large n could be.
2. Karmarkar-Karp (specialized to number partitioning) can find "residues" of size Λ in time $\sim \exp \sqrt{\log \Lambda^{-1}}$.
3. Sieve algorithms (while exponential) are also very efficient and can be generalized past the ADK toy model.

Outline

ADK reduction to number partitioning

The number partitioning problem is:

- given positive integers $\delta_1, \ldots, \delta_n$ to find $s_j \in \{+1, -1\}$ so that

$$\left| \sum_{j=1}^{n} s_j \delta_j \right| \leq 1.$$

Finding ADK vacua is very similar. Define

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

Then

$$\Lambda = \sum_{j=1}^{n} s_j \delta_j.$$

So the numbers involved are real rather than integral.

Random instances of number partitioning

Random instances have been well studied using statistical mechanics.

- set some magnitude parameter B;
- sample n independent numbers $\delta_j \sim \text{Uniform}\{1, 2, \ldots, B\}$;
- define a perfect partition as $s_j = \pm 1$ so that

$$\sum_{j=1}^{n} s_j \delta_j = 0 \text{ if } \sum_{j=1}^{n} \delta_j \text{ even} \qquad \sum_{j=1}^{n} s_j \delta_j = 1 \text{ if } \sum_{j=1}^{n} \delta_j \text{ odd.}$$

Note for a random problem in the limit of large n,

- if $B > 2^{n+O(\log n)}$ will will likely be no perfect partitions,
- if $B < 2^{n+O(\log n)}$ there will be exponentially many partitions.

Note that if $B = \max_j \delta_j$ is only polynomially large, dynamic programming efficiently solves the number partitioning problem.

Cost of brute force search

Consider the number partitioning problem on real numbers.

- An instance is n numbers independently $\sim$ Uniform$[0,1]$.
- It is known that the median optimal residue is $\Theta(\sqrt{n}2^{-n})$.
- Thus, for a solution with residue Λ to exist, one needs $\sqrt{n}2^{-n} < \Lambda$.

This gives problems with

$$n \sim \log_2 \Lambda^{-1} + \frac{1}{2} \log_2 \log_2 \Lambda^{-1} \text{ with}$$
$$b \sim \log_2 \Lambda^{-1} \text{ bits of precision.}$$

Naively, the total complexity of enumeration is $O(nb2^n)$. Instead:

- order tuples $(s_j)_{j=1}^n$ according to a binary reflective Gray code;
- consecutive tuples only differ in one index;
- use the residue from the previous step and add or subtract $2\delta_j$;
- the total complexity of the algorithm is $O(b2^n)$ elementary gates.

This yields a total complexity of order $\Lambda^{-1} \left(\log_2 \Lambda^{-1}\right)^{3/2}$.

Outline

Karmarkar-Karp

The Karmarkar-Karp heuristic is that the largest numbers should be given opposite sign to maximize (relative) cancellation.[10]

- extract the two largest numbers from the list;
- compute their (positive) difference;
- insert the difference back into the list of numbers.

This reduces the problem to a new instance of number partitioning with one fewer number.

The work to perform Karmarkar-Karp goes as

- sorting the initial list has complexity $O(n \log n)$;
- using a heap allows inserting numbers with complexity $O(\log n)$;
- there as exactly $n - 1$ differencing-and-insertion steps.

Thus the total complexity of the algorithm is $O(n \log n)$.

[10] Karmarkar, Karp, *FOCS*, 1982.

Karmarkar-Karp

to find the cosmological constant

Note that if $n \gg \log_2 \Lambda^{-1}$ then $\Lambda^{-1} \ll b2^n$.

- Even the best known exact algorithm scales as $\sim 2^{0.241n}$.

Karmarkar-Karp scales as $n \log(n)$, however will not generally find a perfect partition.

- The K-K residue has smaller size by a factor of $\approx e^{-c \log^2 n}$.
- Asymptotically as $n \to \infty$, we have $c = \frac{1}{\sqrt{2}}$.[11]

For the ADK model, for K-K to find a residue of size Λ

- $n \sim \exp\left[\sqrt{\frac{\log \Lambda^{-1}}{c}}\right]$,

[11] Yakir, *Math. Operations Research*, 1996

Exact algorithms

Number partitioning, subset sum, and knapsack problems are basically the same.

Exact algorithm exists that run in $O(2^{\alpha n})$.

- A straightforward meet-in-the-middle tree search is $\approx 2^{0.5n}$.[12]
- The best known classical algorithm takes $O(2^{0.291n})$.[13]
- The best known quantum algorithm takes $O(2^{0.241n})$.[14]
- The adiabatic algorithm has unknown runtime, but appears to scale as $2^{0.8n}$.[15]

We can use exact algorithms "locally" to produce a sieve heuristic.

- We need to understand the statistics of optimal solutions.

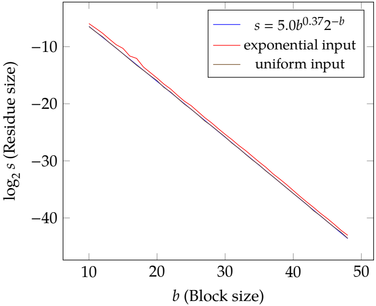
[12] Horowitz, Sahni, *Journ. ACM*, 1974.

[13] Becker, Coron and Joux, *EUROCRYPT*, 2011.

[14] Bernstein, Jeffery, Lange, Meurer, *Post-Quantum Cryptography*, 2013.

[15] Johnson, Aragon, McGeoch, Schevron, *Operations Research*, 1991.

Statistics of the optimal residue (mean)



The optimal residue scales as $\Theta(\sqrt{b}2^b)$.

- Ordinate = mean relative optimal residue size.
- Abscissa = input size for number partitioning problem.
- 1000 instances solved for each $b \in \{10, \ldots, 48\}$.
- Computed with least square error estimator (exponential model).

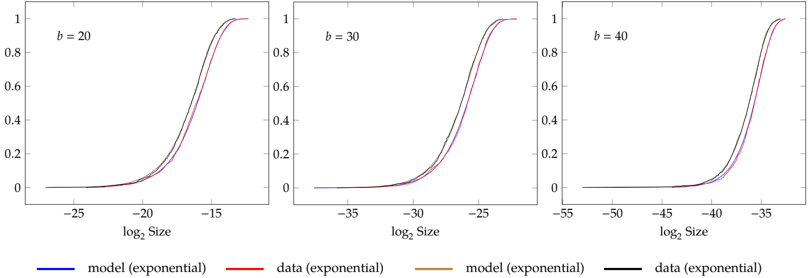
The model $s = 5.0b^{0.37}2^{-b}$ was generated by linear regression.

Statistics of the optimal residue (distribution)

Distribution of the optimal residue for NPP sizes $b = 20, 30, 40$.

- Ordinate = cumulative probability distribution.
- Abscissa = $\log_2$ optimal residue size.
- 1000 data points per plot.

The model is the exponential distribution with mean computed by least squares estimator on data.



A sieve for number partitioning

Here we explore a very simple sieve mechanism.

- ”Lattice sieves” add lattice vectors to produce smaller vectors.
- The simple sieve here is similar in spirit to “tuple” sieve.[16]

In general, a sieve consists of several stages.

- Partition in the input in to small problems.
- Use an exact algorithm to solve the subproblem.
- The results form the input for the next stage of the sieve.

For number partitioning problems,

- we partition all the numbers into blocks of size b,
- we use one the exact methods above, taking work $2^{\alpha b+o(b)}$.

Residues are exponentially distributed with expected size $2^{-b+o(b)}$.

[16] Bai, Laarhoven, Stehlé, *ANTS*, 2016.

A sieve for number partitioning

First stage of the sieve:

Input: n fields of mean energy differences $\delta \approx 1$.

Solve: n/b_1 number partition problems, each of size b_1.

Output: $\frac{n}{b_1}$ residues with mean size $\approx 2^{-b_1}$.

Work: $\approx \frac{n}{b_1} 2^{\alpha b_1}$.

Second stage of the sieve:

Input: n/b_1 residues with mean size $\approx 2^{-b_1}$.

Solve: $n/(b_1 b_2)$ number partition problems, each of size b_2.

Output: $\frac{n}{b_1 b_2}$ residues with mean size $\approx 2^{-(b_1+b_2)}$.

Work: $\approx \frac{n}{b_1 b_2} 2^{\alpha b_2}$.

And so on. After k stages:

Output: single residue of expected length $2^{-t} \approx 2^{-(b_1+\cdots+b_k)}$

Work: $\approx \left(\frac{n}{b_1} 2^{\alpha b_1} + \frac{n}{b_1 b_2} 2^{\alpha b_2} + \cdots + \frac{n}{b_1 b_2 \cdots b_k} 2^{\alpha b_k} \right)$.

A sieve for number partitioning

The optimal work is given by an "equipartition principle:"

- balance the amount of work done on each sieve stage.

Specifically, for stage j and $j-1$ we want

$$\frac{n}{b_1 \cdots b_j} 2^{\alpha b_j} \approx \frac{n}{b_1 \cdots b_{j-1}} 2^{\alpha b_{j-1}} \text{ so we take } b_j - \tfrac{1}{\alpha} \log_2(b_j) \approx b_{j-1}.$$

This table was generated with $\alpha = 0.5$:

k	n	t	$\log_2(\text{Work})$	$(b_1, b_2, \dots)$
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)

(Karmarkar-Karp takes $n \approx 7.8 \times 10^8$ and runs in $w \approx 36.5$.)

Outline

Experiment: Karmarkar-Karp

To empirically test the Karmarkar-Karp algorithm in a regime relevant to the cosmological constant problem within the ADK model:

- one starts with real numbers of order 1, and seeks to find a residue of order $\sim 10^{-120} \approx 2^{400}$;
- scale by a factor of 2^{430} to represented these as integers;
- defined success as achieving residue less than 2^{30}.

The 30 bits of precision deal with "numerical noise."

Therefore an experiment was:

- take n independent $\sim \text{Uniform}\{0, 1, 2, \ldots, 2^{430} - 1\}$,
- test if Karmarkar-Karp achieves a residue less than 2^{30}.

We performed 200 trials for a variety of n around the prediced threshold.

Experiment: Karmarkar-Karp

Predictions

Theory predicts the size of the final residue as exponentially distributed:

- $\Pr\{y < Y < y + dy\} = \lambda e^{-\lambda y} dy$,
- the key parameter is modeled as $\lambda = e^{-c \log^2 n}$,
- asymptotically $c \to 1/\sqrt{2}$ (smaller c are consistently observed).

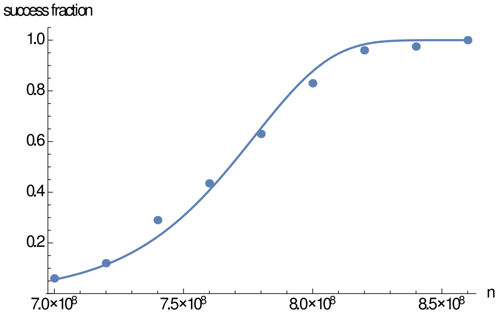
For a reduction factor of $\epsilon = 2^{-400}$, we obtain success probability

$$
\begin{aligned}
P &= \int_0^{\epsilon} \lambda e^{-\lambda y} dy \\
&= 1 - \exp\left[-e^{-c \log^2 n} \epsilon\right].
\end{aligned}
$$

We can use this to fit c to empirical data.

Experiment: Karmarkar-Karp

Numerical results



Plots for a Karmarkar-Karp experiment.

- Ordinate = likelihood in 200 trials of a residue $< 2^{30}$.
- Abscissa = number of samples $\sim$ Uniform$\{0, \ldots, 2^{430} - 1\}$.
- Theory curve $= 1 - \exp\left(-e^{c \log(n)^2} / 2^{400}\right)$.
- Parameter $c = 0.6615$ is fitted. (Asymptotic prediction: $\frac{1}{\sqrt{2}}$.)

Experiment: a toy NPP sieve

As a simple proof of concept, we tackle a toy sieve:

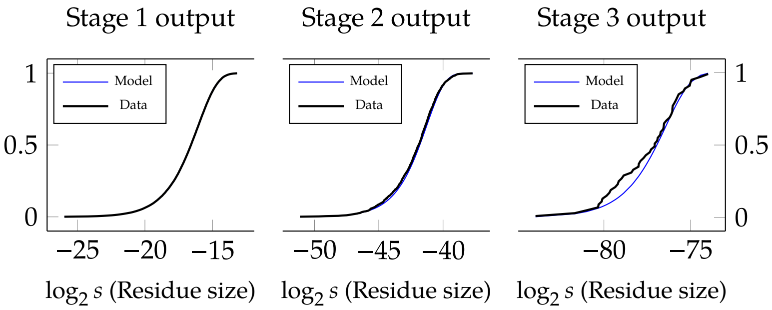
- four stages;
- block sizes $(b_1, b_2, b_3, b_4) = (20, 30, 40, 50)$;
- a simple meet-in-the-middle algorithm ($\alpha = 0.5$).

Stage	b	#Inputs	Distribution	#NPPs	Work	$\mathbb{E}[s]$
One	20	1200000	Uniform	60000	$2^{25.9}$	$2^{-16.1}$
Two	30	60000	Exponential	2000	$2^{26.0}$	$2^{-41.3}$
Three	40	2000	Exponential	50	$2^{25.6}$	$2^{-76.4}$
Four	50	50	Exponential	1	$2^{25.0}$	$2^{-121.3}$

- Work quote is for the entire sieve stage.
- Expected residue size is based on distributional model.

Experiment: a toy NPP sieve

Numerical results



Plots for a four state sieve.

- Ordinate = cumulative likelihood of observing the value.
- Abscissa = (log) size of the optimal residue obtained.
- This final residue was $6.54 \times 10^{-38} \lesssim 2^{-121.3}$.
- The sieve took 152 seconds on my Mac Pro.