

That No Search Is Universal Follows from How the Problems Are Counted

— One Search Is **186.18** Times Faster on One-Humped Functions and Loses at **0.9645** on Structureless Ones
— Only $10^{-113.79}$ of the Averaged Set Has Any Structure — [Paper 324]

Yuuki YAMAGISHI (山岸 勇輝)

Yamagishi Complexity Torus Lab, Japan

August 2026

— *From thinking to syncing.* —

Abstract

The no-free-lunch theorem says that averaged over all problems, every search performs alike. This paper asks what that average is taken over—the answer is a set that contains almost no structure. No new mathematical theorem and no new law is claimed. First, a search that uses structure is fast where structure exists. **186.18** times at $N = 16384$ (Section 2). Second, that speed is logarithmic and random search is linear. $17 \rightarrow 44$ against $32 \rightarrow 8192$ (Section 2). Third, this is the core of the paper. The same search loses to random on structureless functions—**0.9645** (Section 3). Fourth, structure is only $10^{-113.79}$ of the averaged set (Section 4). Fifth, the theorem is correct, and it is a statement about a measure (Section 4). Sixth, the separator is not what is maximised but what is averaged over (Section 5).

Scope of this paper (scope note): No new mathematical theorem and no new law is claimed—the no-free-lunch theorem, the complexity of ternary search, and the counting argument of Kolmogorov complexity are all standard. We do not prove the theorem—this paper only shows its content in numbers, and enters neither its proof nor its generalisations. We do not refute the theorem—the theorem is correct. All this paper says is what its subject is. We do not claim to have measured the balance—what was measured is one pair of algorithms, and the exact accounting across all algorithms and all functions belongs to the theorem, not to this measurement. We do not design search algorithms—genetic algorithms, simulated annealing and Bayesian optimisation are not entered. We claim nothing about the distribution of real problems—which measure real problems follow is not settled here. Relation to earlier papers: Paper 311 showed that the correct bound and the predictive bound are different things—311 is about what is maximised and this paper is about what is averaged over. Objective function and measure: two different places to stand. Paper 186 showed cases where optimising a proxy harder makes the target worse—186 is the gap between target and proxy, this is how the problem set is measured, and both interrogate the premise under which optimisation works. Paper 315 showed which dimension is special depends on the question—this is the same form: which algorithm is better depends on how the problems are measured. Paper 320 showed the bifurcation ratio not depending on the shape of the map—the 186.18 here depends on the shape entirely, the opposite case. Paper 323 showed the three of three primaries coming from the number of receptors—both ask where a number came from. What is added is running ternary and random search at five scales, measuring the sign reversal between the structured and structureless sides, counting the one-humped functions as $10^{-113.79}$ of the whole, and putting the separator on the measure.

1. Notation and the Question

Setting: We search a function assigning a value to each of N points. Evaluations are the budget, and we want the maximum.

Observation 1.1

The no-free-lunch theorem states that averaged uniformly over all functions, every search scores the same. This is correct.

Question: So what is “all functions”?

Observation 1.2

The theorem reads as saying something about algorithms, and it is in fact saying something about a set of problems—which of the two is the subject changes what it means.

2. Using Structure Is Fast Where Structure Exists

Setting: Apply ternary search to a one-humped (unimodal) function. Two probes tell which side the peak is on, so the bracket shrinks.

Check 2.1: Evaluations needed to reach the exact maximum (computed here, 4000 trials each).

N	Ternary	Random ($N/2$)	Ratio
64	17.000	32.0	1.88
256	24.000	128.0	5.33
1024	30.000	512.0	17.07
4096	37.000	2048.0	55.35
16384	44.000	8192.0	186.18

Observation 2.2

Ternary search reached the exact optimum in 100.00% of trials. It misses at no scale—given that the function is known to be unimodal, this is a guarantee.

Observation 2.3

They grow differently. Ternary goes $17 \rightarrow 44$ (logarithmic), random goes $32 \rightarrow 8192$ (linear)—a 256-fold increase in scale opened the ratio from 1.88 to 186.18.

Observation 2.4

The ratio grows without bound. The same shape as Paper 303's "if the opponent is an exponential, write not how many times but at which n they cross"—here there is no crossing; it only keeps opening.

3. The Core: The Same Search Loses Where There Is No Structure

Setting: Now assign the values uniformly at random. The same ternary search, the same budget.

Check 3.1: Mean of the best value found (20000 trials each).

N	Budget	Ternary	Random	Ratio
64	23	58.6623	60.8210	0.9645
256	30	242.2745	247.2238	0.9800
1024	37	984.5823	996.7134	0.9878

Observation 3.2

This is the core of the paper. The sign has reversed—on the structured side it wins by 186.18, on the structureless side it loses at 0.9645. It is the same algorithm.

Observation 3.3

The reason it loses is visible. Ternary search spends evaluations narrowing a bracket, and when the values are random that bracket means nothing—every narrowing adds territory it will never look at.

Observation 3.4

This is the no-free-lunch balance made visible. The gain on structure is paid for somewhere—and the theorem says the account is exactly zero.

Observation 3.5

A point to write honestly——this paper did not measure the balance.What was measured is one pair of algorithms, while the exact account across all algorithms and functions belongs to the theorem——what is shown here is the direction, not the amount.

4. The Averaged Set Contains Almost No Structure

Check 4.1: The fraction of N -point functions that are unimodal.

N	All functions	Unimodal ones	Fraction
64	$10^{115.60}$	64	$10^{-113.79}$
256	$10^{616.51}$	256	$10^{-614.10}$

Observation 4.2

It is $10^{-113.79}$.Far smaller than the reciprocal of the number of atoms in the universe——draw one uniformly and being unimodal effectively never happens.

Check 4.3: The same holds for far weaker senses of structure.Writing one function on $N = 64$ takes 384 bits.

Compressible by k bits	Fraction of such functions (at most)
8	3.9062×10^{-3}
32	2.3283×10^{-10}
128	2.9387×10^{-39}
256	8.6362×10^{-78}

Observation 4.4

At most 2^{-k} of objects can be compressed by k bits. This follows from counting alone—there are only 2^{n-k} short descriptions, and no more will fit.

Observation 4.5

“Having structure” means “being writable briefly”. And things writable briefly are, by definition, almost absent—under the uniform measure, structure is close to measure zero.

Observation 4.6

The theorem is correct, and it is a statement about a measure. Averaged uniformly, every algorithm is alike—and what dominates that average is functions nobody has ever posed.

5. Separator: Not What Is Maximised but What Is Averaged Over

Observation 5.1

Separator—which measure is put on the set of problems.

Measure	Conclusion	Correct?
uniform	every search alike	yes
weighted to structure	structure-using search wins	yes

Observation 5.2

The two rows do not contradict. The same algorithms, functions and budget, with only the measure changed—the difference in conclusion comes not from a property of the algorithm but from how the question was posed.

Observation 5.3

This is where this paper parts from Paper 311.

What changes the conclusion		Category
Paper 311	what is maximised	objective function
This paper	what is averaged over	measure

Observation 5.4

311 set two objectives side by side, efficiency and power. This paper holds the objective fixed and changes the set of problems——by Paper 300’s criteria they have different roots and no translation can be constructed.

Observation 5.5

Paper 300 wrote that agreement of values alone is not sameness of root. The reversal here obeys the same discipline——the reason 186.18 and 0.9645 come from one algorithm is nameable, as whether the bracket means anything.

Observation 5.6

A point to write honestly——this paper does not settle which measure real problems follow. That it is not uniform can be shown; what it is cannot——that belongs to experience.

Observation 5.7

One last thing——“no free lunch” is not about the canteen but about the menu. Only $10^{-113.79}$ of the dishes the theorem counts are ones anyone has ever eaten——and what we are actually paying for is the dishes that are not on it.

6. Roll Call and Closing

1. **186.18 at $N = 16384$ (Check 2.1): If unimodal.**
2. **Exact optimum 100.00% of the time (Observation 2.2): Ternary does not miss.**
3. **$17 \rightarrow 44$ against $32 \rightarrow 8192$ (Observation 2.3): Logarithmic and linear.**
4. **0.9645 without structure (Check 3.1): The same algorithm loses.**
5. **Because the bracket means nothing (Observation 3.3): Narrowing costs.**
6. **The direction, not the amount (Observation 3.5): The account is the theorem's.**
7. **Unimodal is $10^{-113.79}$ (Check 4.1): At $N = 64$.**
8. **k bits compress at most 2^{-k} (Observation 4.4): From counting alone.**
9. **Only the measure changed (Observation 5.2): Not a property of the algorithm.**
10. **A different category from 311 (Observation 5.3): Objective against measure.**

Closing: The no-free-lunch theorem says that averaged over all problems, every search performs alike. The theorem is correct—what this paper asks is what that average is taken over, and the answer is a set containing almost no structure. Ternary search on a one-humped function is **186.18** times faster at $N = 16384$ and reaches the exact optimum in **100.00%** of trials—its evaluations grow $17 \rightarrow 44$, logarithmically, against random search's $32 \rightarrow 8192$, linearly, so a 256-fold increase in scale opens the ratio from 1.88 to 186.18, and it grows without bound. And yet the same ternary search, on functions whose values are random, loses to random search—at **0.9645**, the sign reversed. It spends evaluations narrowing a bracket, and that bracket means nothing. This is the no-free-lunch balance made visible—though what was measured is one pair of algorithms, and the exact account belongs to the theorem. What is shown is the direction, not the amount. Counting the side being averaged over, unimodal functions are $10^{-113.79}$ of the whole at $N = 64$ —and the same holds for far weaker structure, since at most 2^{-k} of objects compress by k bits, which follows from counting alone. “Having structure” means “being writable briefly”, and things writable briefly are by definition almost absent. One thing separates them—which measure is put on the set of problems. Uniform gives “every search alike” and it is correct; weighted to structure gives “the structure-using search wins” and it is equally correct. The two do not contradict, and the difference comes not from a property of the algorithm but from how the question was posed. This is a different category from Paper 311's “the correct bound and the predictive bound are different things”—311 is what is maximised, this is what is averaged over, and objective function and measure have different roots with no translation between them. One last thing—“no free lunch” is not about the canteen but about the menu. Only $10^{-113.79}$ of the dishes the theorem counts are ones anyone has ever eaten, and what we are actually paying for is the dishes that are not on it.

References

1. Wolpert, D. H. and Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1).
2. Schaffer, C. (1994). A Conservation Law for Generalization Performance. *Proceedings of the Eleventh International Conference on Machine Learning*.
3. Li, M. and Vitányi, P. (2019). *An Introduction to Kolmogorov Complexity and Its Applications*, 4th edition. Cham: Springer.
4. Igel, C. and Toussaint, M. (2005). A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions. *Journal of Mathematical Modelling and Algorithms*, 3(4).
5. Yamagishi, Y. (2026). *When a Proxy Becomes the Target It Stops Being a Proxy* [Paper 186]. Zenodo. 10.5281/zenodo.22105530
6. Yamagishi, Y. (2026). *The Three Hundredth Census: Whether Two Things Share a Root Is Decidable* [Paper 300]. Zenodo. 10.5281/zenodo.22171914
7. Yamagishi, Y. (2026). *The Formula Returns 4, and at That One Point the Proof Does Not Reach* [Paper 303]. Zenodo. 10.5281/zenodo.22172026
8. Yamagishi, Y. (2026). *The Correct Bound Is Not the Predictive One* [Paper 311]. Zenodo. 10.5281/zenodo.22174391
9. Yamagishi, Y. (2026). *Which Dimension Is Special Depends on the Question* [Paper 315]. Zenodo. 10.5281/zenodo.22208139
10. Yamagishi, Y. (2026). *The Road into Chaos Carries a Number the Map Does Not* [Paper 320]. Zenodo. 10.5281/zenodo.22209089
11. Yamagishi, Y. (2026). *The “Three” in Three Primaries Is Not a Property of Light* [Paper 323]. Zenodo. 10.5281/zenodo.22209319

— *From thinking to syncing.* —

Yuuki YAMAGISHI (山岸 勇輝)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)