

The Asymptotic Efficiency of Bounded Search

Status: Working Paper | Date: 2026-04-01 | Status: KEGA Synthesis Complete

The Asymptotic Efficiency of Bounded Search: An Economic Framework for Automated Mathematical Discovery

Status: Working paper | Date: 2026-04-01 | Method: Heuristic Bounding and Search-Space Reduction

Abstract

We propose a formal economic framework for mathematical problem-solving, treating discovery as a production process within a computational factory. By shifting the role of Artificial Intelligence from a direct "solver" to a "heuristic bound-finder," we demonstrate that even marginal gains in search-space reduction yield superlinear returns as time approaches infinity (**T -> infinity**). We define the "Compounding Search Engine," where identifying patterns in mathematical structures leads to tighter operational bounds, which in turn accelerates subsequent discovery. The model suggests that the human role in this system transitions from manual calculation to high-level "stall detection" and system redirection.

Prequel: Foundational Concepts

Before diving into the core thesis, understand these 12 foundational concepts that bridge mathematics, algorithmic search, and economics.

- Solution Space (The Map):** Imagine every possible answer to a problem as a point on a giant map. For complex problems, this map is vast and mostly filled with incorrect answers.
- Search Complexity (The Haystack):** This is the measure of how difficult it is to find the "needle" (the solution) in the solution space. As problems get harder, the haystack grows exponentially.
- Bounding (Fencing Off):** Bounding is the act of identifying regions of the solution space where the solution *cannot* exist. By "fencing off" these areas, you focus only on the relevant search area.
- Heuristic Principles (Rules of Strategy):** Instead of rigid axioms, these are strategic "rules of thumb" that help us decide where to look first. They aren't always perfect, but they save time.
- Operational Bounding:** Applying bounding as an active strategy during a calculation. For example, knowing that a physical variable must be positive immediately discards 50% of the possible numerical range.
- AI as a Pattern Scout:** Rather than asking AI to solve the whole problem, we use it to "scout" for patterns that suggest where a new boundary or constraint might lie.
- Efficiency Gain:** The percentage of the search space that is removed by a new boundary. Removing

87.5% of a 3D search space by knowing a single vector is a massive efficiency gain.

8. **Cost of Discovery (Ct)**: The upfront investment of compute and time required to find a new boundary or heuristic. This is a "sunk cost" in the discovery process.

9. **Search Reduction (Gt)**: The cumulative gain in speed and compute saved every time a discovered boundary is applied to future problems.

10. **Compounding Loops (The Feedback Engine)**: Every successful search provides more data for the "Pattern Scout," allowing it to find even tighter boundaries, which in turn makes the next search even faster.

11. **Economic Factory of Discovery**: Treating the research process like a manufacturing line. The "factory" takes in raw compute and outputs verified mathematical or scientific discoveries with increasing efficiency.

12. **Stall Detection (The Roomba Kick)**: The human role in this system. Humans monitor the "factory" and intervene (or "kick the Roomba") only when the AI gets stuck or the efficiency gains plateau.

The Asymptotic Efficiency of Bounded Search

1. Introduction

Traditional mathematical discovery often relies on brute-force search or human intuition--a process that is both costly and difficult to scale. While Artificial Intelligence (AI) has shown promise in solving specific sub-problems, its most significant potential lies in **dimensionality reduction** and **search-space bounding**.

This paper establishes a unifying view: mathematical reasoning and algorithmic search are both forms of navigating a high-dimensional solution space. By applying heuristic principles to "fence off" irrelevant areas of this space, we can achieve massive efficiency gains. Following the work of **Terence Tao (2025)** on AI-assisted mathematics, we propose a "tighter bound" on AI's utility: its primary value is as an accelerator that shapes the problem for the human/AI solver.

2. The Mechanics of Bounded Search

2.1 Heuristic Principles of Reduction

Rather than treating math solutions as absolute axioms, we treat them as **operational strategies**. For any given problem (e.g., an Ordinary Differential Equation or a 3D search problem), known constraints can reduce the domain or range.

* **Case Example (3D Search)**: If the search direction is known via a vector, the search space is immediately reduced to a single octant (1/8th of the original space), achieving an **87.5% efficiency gain**.

2.2 The Role of Pattern Recognition

Machine Learning (ML) acts as a **"Pattern Scout."** Its role is to identify where boundaries likely exist within the computational landscape before the heavy compute of a full search is triggered.

3. The Asymptotic Efficiency Model

3.1 The Theorem of Bounded Gains

The core of our efficiency model relies on the **Asymptotic Efficiency Theorem** (see: [Original Theorem Post](http://149.28.225.2/posts/asymptotic-efficiency-theorem/)). We define **C** as the fixed cost of discovering a boundary and **G** as the efficiency gain (the factor by which the search space is reduced).

- * **Key Formula:** The discovery of a boundary is economically viable if the cumulative savings over **T** iterations exceed the cost **C**.
- * **The Multiplier:** Because finding a boundary is a fixed upfront cost while its application provides a recurring benefit, the "Efficiency Multiplier" approaches infinity as **T** -> **infinity**. Even a marginal 1% gain (**G = 1.01**) eventually offsets any finite discovery cost.

3.2 The Compounding Search Engine

We define the **Compounding Search Engine** as a closed feedback loop where each discovered bound (output) becomes high-fidelity training data that lowers the cost **C_{t+1}** for the next discovery.

- * **Logical Implication:** Because the cost of discovery **C_t** decreases via learning while the gains **G_t** stay positive, the net efficiency curve becomes superlinear.
- * **Layman Insight:** It's like a factory that gets faster every time it ships a product--the more math it solves, the cheaper the next solution becomes. Eventually, the factory effectively runs itself.

4. The Economic Factory of Discovery

We propose a formal **Ontology of Math Goods**, where a solved theorem or a validated search result is treated as a manufactured commodity. In this model, computational biology or space-search problems are not "riddles" but "production units" moving through a factory line.

The **Singularity Factory** is the ultimate expression of this ontology: a system where the AI discovers a new bound, uses it to solve a problem, extracts a new pattern from that solution, and generates a tighter bound--all in a millisecond loop. This "200 MPH highway" represents the transition from linear human-led research to exponential automated discovery.

5. Conclusion: The Human as Roomba-Kicker

In this high-speed "Discovery Factory," the human role is refined to system-level maintenance. We call this "**Stall Detection**." Just as a user might kick a stuck Roomba, the human mathematician intervenes when the AI's pattern recognition hits a logic wall or when search-space reduction falls below a critical threshold. The human provides the "out-of-domain" kick that redirects the factory back onto a productive path.

Appendix: 80/20 Gap Analysis (KEGA)

Target: 80% efficiency gain with 20% effort

1. The Master Gap: The "Unit of Bounding" Problem

We lack a formal metric to quantify how much "computational energy" is saved per "bit of heuristic constraint."

* **The 80/20 Solution:** Develop a "**Bounding ROI**" (**bROI**) formula where **bROI = (Search Space Reduction %) / (Compute Cost of Discovery)**. Identifying high-bROI constraints is the 20% work that yields 80% of the factory's output.

2. The Implementation Gap: Pattern-to-Axiom Translation

Machine Learning outputs "fuzzy" patterns (probability heatmaps), while solvers require "rigid" constraints.

* **The 80/20 Solution:** Create a "**Soft-to-Hard**" **Logic Bridge**--an automated system that takes ML "hints" and uses a symbolic prover (like Z3) to verify if they can be hardened into a formal mathematical bound.

3. The Human Gap: The "Stall Detection" Metric

When exactly should the human "kick the Roomba"?

* **The 80/20 Solution:** Define a "**Discovery Velocity**" threshold. If the search space is not reducing by at least 1% per **T** iterations, the system triggers an automatic Human-in-the-Loop (HITL) request, preventing 80% of wasted compute.

QED

References

1. **Tao, T.** (2025). *AI-Assisted Mathematics and the Future of Formal Proofs*. [Available Online](<https://terrytao.wordpress.com/>).
2. **David.** (2026). *Asymptotic Efficiency Theorem*. [Original Thesis](<http://149.28.225.2/posts/asymptotic-efficiency-theorem/>).