THE YANG UNIFIED CONSTRUCTIVE PRIME GENERATOR: TOWARD AN OPTIMAL HYBRID SIEVE

PU JUSTIN SCARFY YANG

Abstract. We present the Yang Dynamic Generative Sieve (YDGS), a novel constructive algorithm for generating prime numbers incrementally. Unlike classical sieving techniques such as the Sieve of Eratosthenes or the segmented priority queue sieve, YDGS defines primality as a dynamic process of logical filtration from previously known primes. We formalize its mathematical structure, prove its soundness and completeness, and propose a hybridized expansion—incorporating wheel factorization, priority queues, and AI-guided heuristics—to initiate the construction of a universal, optimally efficient prime generation framework.

Contents

1. Introduction

The search for primes is foundational in number theory, algorithmic arithmetic, and modern cryptography. While classical sieve methods are efficient for bounded domains, they lack adaptability for infinite or dynamic contexts. In this paper, we formalize a purely constructive and logically generative approach—termed the **Yang Dynamic Generative Sieve (YDGS)**—and explore its expansion into a hybrid architecture for producing primes with theoretical and practical efficiency.

Date: May 15, 2025.

2. Review of Classical Prime Generation

2.1. **Eratosthenes Sieve.** A bounded elimination sieve with time complexity $O(n \log \log n)$, requiring $O(n)$ space. Non-incremental and non-streamable.

2.2. **Wheel Sieve.** Introduces pre-filters based on small primes (e.g., 2, 3, 5), allowing strides to skip known composite classes.

2.3. **Priority Queue Sieve.** Uses a min-heap or tree to track the next multiple of each known prime. Supports incremental generation but can be memory-intensive.

2.4. **Lazy Sieve of O'Neill.** A Haskell implementation of lazy evaluation via functional programming, aesthetically elegant but not optimal for large-scale prime streams.

2.5. **AKS Primality Test.** A deterministic primality verifier rather than a generator, with complexity $O(\log^6 n)$. Not used for enumerating primes.

3. The Yang Dynamic Generative Sieve (YDGS)

Definition 1 (Dynamic Constructive Sieve)**.** *Let $\mathcal{P}_n = \{p_1 = 2, p_2 = 3, \ldots, p_n\}$ be the known primes. Define the next candidate set:*

$$S_n := \left\{ m > p_n \mid \forall\, p_i \in \mathcal{P}_n,\, p_i^2 \leq m \Rightarrow p_i \nmid m \right\}.$$

Then define:

$$p_{n+1} := \min S_n.$$

Theorem 1 (Soundness)**.** *Each p_{n+1} selected by the algorithm is a prime number.*

Proof. If p_{n+1} were composite, it would have a factor $a \leq \sqrt{p_{n+1}}$, contradicting the filter condition. Thus, p_{n+1} must be prime. □

Theorem 2 (Completeness)**.** *Every prime number appears in the sequence generated by YDGS.*

Proof. Suppose a prime q is omitted. All primes less than q appear and thus will be used to test q, which would pass the test and be selected. Contradiction. □

4. Toward a Hybrid Optimal Generator

We now define an extension of YDGS—called the **Yang Unified Constructive Prime Generator (YUCPG)**—as a synthesis of multiple paradigms:

- **Base: YDGS** as the logical core.
- **Wheel Factorization:** Pre-filters by residue classes modulo the product of small primes.
- **Priority Queue:** Efficient composite scheduling via heap structure.
- **Segmented Memory:** Cache-aware memory windows to reduce latency.
- **Heuristic Jump Table:** AI-guided probabilistic filters to skip low-density zones.

Remark 1. *Such hybridization aims to maintain mathematical clarity while enabling high-performance real-world usage. Each module is detachable, allowing future refinements.*

5. Philosophical Position and Future Directions

The YUCPG is not merely an algorithm—it is a philosophy of construction. Rather than treating primes as accidents discovered by elimination, it defines them as logical inevitabilities under structured expansion. It suggests new directions:

- Generalizing primality within non-commutative or Yang_n number systems;
- Exploring links with category-theoretic generation and homotopy-theoretic logic;
- Embedding the framework into proof assistants (e.g., Coq, Lean) for certified mathematics.

6. Conclusion

We have introduced a new constructive sieve approach (YDGS) and proposed a unifying framework (YUCPG) that blends traditional algorithms with formal mathematical clarity. This structure provides fertile ground for both practical optimization and theoretical generalization in prime generation research.

References

[1] Melissa O'Neill, *The Genuine Sieve of Eratosthenes*, Journal of Functional Programming, 15(4): 415–433, 2004.
[2] Richard Bird, *Pearls of Functional Algorithm Design*, Cambridge University Press, 2010.
[3] Richard Crandall and Carl Pomerance, *Prime Numbers: A Computational Perspective*, Springer, 2005.
[4] Donald E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed., Addison-Wesley, 1997.