

A Logarithmic Adaptive Algorithm for Constructing Goldbach Prime Pairs in Very Large Even Integers

Bouchaib Bahbouhi* 

Independent Researcher, Nantes, France

*Corresponding Author

Bouchaib Bahbouhi, Independent Researcher, Nantes, France.

Submitted: 2026, Sep 09; Accepted: 2026, Oct 19; Published: 2026, Oct 28

Citation: Bahbouhi, B. (2026). A Logarithmic Adaptive Algorithm for Constructing Goldbach Prime Pairs in Very Large Even Integers. *Int Nat Sci Int Rese*, 1(2), 01-28.

Abstract

The constructive search for Goldbach representations of very large even integers becomes increasingly difficult when direct primality testing is applied indiscriminately to candidate pairs. This work presents an adaptive computational framework designed to reduce that cost by combining centered logarithmic windows, peripheral searches, deep modular sieving, survivor ranking, adaptive zone switching, and strong probable-prime testing. For an even integer E , the algorithm uses both the centered parametrization $p = E/2 - d$, $q = E/2 + d$ and the peripheral relation $q = E - p$. Three logarithmic scales, $H_2 = (\log E)^2$, $H_3 = (\log E)^3$, and $H_\infty = (\log E)^3 \log \log E$, are used as search and diagnostic scales rather than as proved universal Goldbach bounds.

The central principle of the method is MAX-SIEVE FIRST: candidates are filtered at the deepest practical sieve level before expensive primality tests are applied. Surviving candidates are ranked by resistance to progressively deeper sieving, and the algorithm dynamically switches between central, square-root, and peripheral regions when long composite-composite chains or local deserts are detected. A diagnostic SURGERY mode is introduced for difficult cases in order to distinguish low admissible density from failures of simultaneous primality synchronization.

The method was tested on even integers containing from hundreds to several thousand decimal digits. Prime-prime and probable-prime Goldbach representations were obtained in both centered and peripheral regimes, including certified examples around 10^{3000} and a peripheral representation for $E = 10^{6000}$. The experiments indicate that the algorithm remains computationally efficient up to approximately 10^{2000} digits, begins to slow significantly in the 10^{2500} – 10^{3000} range, and becomes increasingly dominated by large-integer primality testing beyond approximately 10^{3000} digits. These thresholds are empirical rather than mathematical limits. The results support a computational distinction between modular admissibility and bilateral prime synchronization and provide a practical framework for large-scale Goldbach pair construction.

Keywords: Goldbach Conjecture, Goldbach Pairs, Prime-Pair Search, Logarithmic Window, Adaptive Sieve, Centered Search, Peripheral Search, Primality Testing, Goldbach Desert, Computational Number Theory

1. Introduction

Goldbach's binary conjecture, stating that every even integer greater than 2 can be expressed as the sum of two primes, is among the oldest unresolved problems in additive number theory. Its modern formulation originates from the correspondence between Christian Goldbach and Leonhard Euler in 1742. Despite enormous theoretical and computational progress, no proof valid

for every even integer is currently known.

The analytical study of Goldbach-type problems played a central role in the development of the circle method. Hardy and Littlewood introduced a systematic framework for additive prime problems and formulated quantitative conjectures for the expected number of representations of large integers as sums of primes

[1,2]. Vinogradov later established the ternary Goldbach theorem for all sufficiently large odd integers, and the modern completion of the ternary problem by Helfgott proved that every odd integer greater than 5 is the sum of three primes [3-5]. Helfgott's proof combines the circle method, large-sieve techniques, and explicit exponential-sum estimates.

For the binary problem, sieve methods provide some of the strongest unconditional approximations to Goldbach. Chen's theorem shows, in particular, that every sufficiently large even integer can be written as the sum of a prime and an integer having at most two prime factors. This result represents one of the closest classical approaches to the binary conjecture and illustrates both the power and the intrinsic parity limitations of sieve methods [6-8]. Modern work related to Chen-type arguments continues to refine almost-prime results and their interaction with sieve theory.

Another major line of progress concerns exceptional sets. Instead of proving the conjecture for every even integer, one studies how many integers may fail to possess a Goldbach representation. Montgomery and Vaughan obtained influential bounds for the exceptional set of Goldbach's problem, contributing to the broader conclusion that the conjecture holds for almost all even integers in appropriate asymptotic senses [9].

Computational verification constitutes a complementary direction. Large-scale calculations have confirmed the binary Goldbach conjecture throughout enormous finite intervals. Oliveira e Silva, Herzog, and Pardi verified the conjecture for every even integer up to 4×10^{18} and simultaneously collected extensive information on minimal Goldbach partitions and prime gaps [10]. Their results also showed strong agreement between empirical counts and Hardy–Littlewood-type predictions.

However, exhaustive verification and constructive search for isolated extremely large integers are fundamentally different computational tasks. Exhaustive verification exploits ordered traversal, sieving over large intervals, and reuse of arithmetic information. By contrast, if a single even integer E has several thousand decimal digits, the main practical problem is to identify a promising pair p and $q = E - p$ without performing expensive primality tests on a very large number of hopeless candidates.

The cost of primality testing is therefore central to the present problem. Miller introduced an important primality test whose deterministic form depends on a generalized Riemann-hypothesis assumption, while Rabin transformed the method into an unconditional probabilistic test [11,12]. Repeated Miller–Rabin testing can reduce the error probability exponentially and remains one of the most widely used practical tools for screening large prime candidates. Probable-prime testing must nevertheless be distinguished from primality proof.

For certified primality, elliptic-curve methods are particularly important. Atkin and Morain developed the elliptic curve primality

proving method, ECPP, into an effective practical algorithm capable of producing primality certificates for very large integers [13]. The emergence of deterministic polynomial-time primality testing through the work of Agrawal, Kayal, and Saxena further established that primality belongs to the complexity class P , although AKS is generally not the fastest method for the very large practical computations considered here [14]. Modern computational treatments of primes, probable primes, certification, and fast large-integer arithmetic are developed extensively by Crandall and Pomerance [15].

The present work addresses a different computational question from exhaustive Goldbach verification. Given one very large even integer E , sometimes containing thousands of digits, how can a Goldbach pair be found while minimizing the number of expensive primality tests?

The method developed here starts from two complementary parametrizations. In the centered representation, $p = E/2 - d$, $q = E/2 + d$, so the search is performed through deviations d from the midpoint. In the peripheral representation, $q = E - p$, and small or moderate primes p are examined while q remains close to E .

The centered formulation naturally leads to logarithmic search scales. Three scales are considered throughout this work: $H_2(E) = (\log E)^2$, $H_3(E) = (\log E)^3$, and $H_\infty(E) = (\log E)^3 \log \log E$.

These quantities are used as heuristic window sizes and diagnostic scales. No claim is made here that every even integer necessarily possesses a Goldbach representation inside any one of these windows.

A principal innovation of the computational strategy is the separation of modular admissibility from actual primality. For each small prime r , residue classes forcing divisibility of either member of the candidate pair can be removed before any strong primality test is performed. This generates a deterministic set of admissible survivors. In practice, a large fraction of candidate pairs disappears at this stage.

The algorithm therefore adopts a MAX-SIEVE FIRST strategy. Rather than applying a weak sieve and immediately testing numerous large integers for probable primality, the deepest practically affordable sieve is applied first. Only the most resistant survivors are passed to Miller–Rabin or related strong probable-prime tests. The expensive part of the computation is consequently concentrated on a relatively small premium subset.

A second feature is adaptive spatial search. The algorithm does not assume that the most useful Goldbach representation must always lie close to the midpoint. It can operate in the central region, near a square-root-type scale, or in the peripheral region. When a region produces a long sequence of composite-composite outcomes, denoted C/C , the procedure identifies the region as locally cold and can move to another search zone instead of continuing a brute-

force scan.

For particularly difficult cases, a diagnostic procedure called Surgery is used. The objective is to determine whether the observed difficulty originates from low modular admissibility, unusually large gaps between survivors, asymmetric behavior between the two sides of the pair, or a more subtle failure of simultaneous primality. This distinction becomes important because several of the experiments reported below show that a region may contain many highly admissible candidates while still producing no prime-prime event for a long interval.

This observation motivates the concept of a Goldbach synchronization desert. Such a desert is not simply a region devoid of candidates that survive small-prime sieving. It is a region in which admissible candidates continue to occur but simultaneous primality of p and q fails repeatedly. In the experimental framework used here, this effect appears to become increasingly important in difficult high-digit cases.

The algorithm has been applied to numerous even integers ranging from moderate numerical sizes to several thousand decimal digits. Both centered and peripheral searches were investigated. Among the resulting examples are probable-prime and certified prime-prime representations for even integers near 10^{1000} , 10^{2000} , 10^{2500} , 10^{2600} , 10^{2750} , 10^{3000} , 10^{4000} , and 10^{6000} . Detailed examples and their verification status are provided in the Results section and in Appendix 1.

The experiments also reveal a practical transition in computational behavior. For the present implementation, searches up to roughly 10^{2000} digits remain relatively comfortable. Between approximately 10^{2000} and 10^{3000} digits, the cost of strong probable-prime testing becomes increasingly visible. Beyond roughly 10^{3000} digits, the sieve itself can still identify promising survivors efficiently, but repeated strong primality tests on numbers containing several thousand digits become the dominant cost. This is an empirical computational transition rather than a theoretical complexity boundary.

The aims of this paper are therefore fourfold. First, to describe a practical adaptive algorithm for constructing Goldbach pairs in isolated very large even integers. Second, to quantify the benefit of deep preliminary sieving and survivor ranking. Third, to distinguish modular admissibility deserts from bilateral prime-synchronization deserts. Fourth, to document a collection of large explicit Goldbach representations obtained with the method. The work does not claim a proof of the binary Goldbach conjecture. A separate appendix presents the strongest theoretical reduction obtained in the associated research program, including the remaining uniform Möbius-cancellation condition. That reduction is included to explain the relationship between the computational observations and the unresolved analytical obstruction, but it is explicitly separated from the proved and experimentally verified content of the algorithmic study.

1.1 Algorithmic Framework

The computational problem considered in this work is the following: given a very large even integer E , find two primes p and q such that $p + q = E$.

The objective is not to enumerate all Goldbach representations of E , but to identify one reliable pair while minimizing expensive primality tests. The algorithm therefore separates candidate generation, modular admissibility, survivor ranking, strong probable-prime testing, and final certification.

1.1.1 Basic Notation

Let E be an even integer and define $m = E/2$.

Two complementary representations are used.

• Centered Representation

A candidate pair is written as $p = m - d$, $q = m + d$, where d is a non-negative integer deviation from the midpoint. The identity $p + q = E$ is then automatic.

This representation is particularly useful for studying local Goldbach behavior near $E/2$, for measuring central deserts, and for comparing the first successful deviation with logarithmic scales.

• Peripheral Representation

Alternatively, one chooses a relatively small or moderate prime p and defines $q = E - p$.

The peripheral mode is especially useful when a representation exists with a small first prime. It avoids searching through a large central interval and can produce very rapid successes even for extremely large E .

The method deliberately retains both representations because experiments show that a difficult central case can sometimes possess an easy peripheral representation.

1.2 Logarithmic Search Scales

Three logarithmic scales are used throughout the computational study.

The first is $H_2(E) = (\log E)^2$.

This is treated as a critical heuristic scale. In probabilistic models of prime pairs, a central interval of this approximate order is naturally associated with an expected number of Goldbach candidates of constant order. The second scale is $H_3(E) = (\log E)^3$.

This is treated as a supercritical practical scale. Relative to H_2 , it offers an additional logarithmic factor and therefore contains substantially more opportunities for simultaneous primality.

The third scale is

$$H_\infty(E) = (\log E)^3 \log \log E.$$

This scale provides an additional slowly increasing safety margin and is useful when difficult or adversarial cases are examined.

These scales are not asserted to be universal Goldbach bounds. They are algorithmic search windows and diagnostic quantities.

For a successful centered pair, the minimum observed deviation is denoted $d_{\min}$. The normalized quantities $\rho_2 = d_{\min} / H_2(E)$,

$\rho_3 = d_{\min} / H_3(E)$, and $\rho_{\infty} = d_{\min} / H_{\infty}(E)$ are useful for comparing cases of very different numerical sizes.

1.3 Modular Admissibility

Before any strong primality test is performed, candidates are filtered using small primes. For a centered pair, $p = m - d$, $q = m + d$.

For each small prime r , values of d satisfying $m - d \equiv 0 \pmod{r}$ or $m + d \equiv 0 \pmod{r}$ are rejected, except when the corresponding candidate itself equals r .

Thus, for most small primes r , two residue classes of d modulo r are forbidden.

The set of deviations surviving all selected modular tests is called the admissible deviation set.

This step is completely deterministic and independent of probable-prime testing.

In peripheral mode, the same principle is applied directly to $q = E - p$. If a small prime r divides q , then p satisfies $p \equiv E \pmod{r}$.

Candidates falling into such residue classes can therefore be removed before q is constructed in full precision or subjected to a strong primality test.

1.4 Survivors

A candidate is called a survivor at sieve depth B if neither member of the pair has been eliminated by any prime $r \leq B$ used in the sieve.

Survivors should not be confused with probable primes.

A survivor only satisfies the statement:

no tested small prime divides the candidate pair.

This distinction is essential for large-scale experiments.

For example, a candidate can survive all primes below 100,000 or 1,000,000 and still be composite because of a much larger factor.

The algorithm therefore records three different statuses: admissible, probable prime, certified prime.

These categories are never merged.

1.5 Max-Sieve First

The central operational principle of the algorithm is MAX-SIEVE FIRST.

Instead of using a shallow sieve and immediately applying expensive primality tests to many candidates, the procedure attempts to use the deepest reasonably computable sieve before strong testing.

The workflow is therefore raw candidates $\rightarrow$ modular filtering

$\rightarrow$ deep sieve survivors

$\rightarrow$ premium survivors

$\rightarrow$ strong probable-prime testing $\rightarrow$ certification.

The motivation is straightforward.

For an integer containing several thousand digits, trial elimination by a small prime is extremely cheap compared with a Miller–Rabin, Baillie–PSW, or ECPP-type operation.

A deeper preliminary sieve can therefore save a large number of costly tests.

1.6 Survivor Resistance

Not all survivors are treated equally.

Let $B_1 < B_2 < \dots < B_k$ denote increasing sieve depths.

A candidate surviving only to B_1 has low resistance.

A candidate surviving every level up to B_k has high resistance.

The algorithm assigns higher priority to candidates that remain admissible under progressively deeper sieving.

This concept is referred to as survivor resistance.

In practice, premium survivors are those that remain after the maximum sieve depth currently considered practical.

The strongest primality tests are concentrated on this premium subset.

1.7 First-Hit Mode

For many computational applications, the objective is simply to find one Goldbach representation.

In this case the algorithm uses first-hit mode.

The procedure terminates immediately after the first candidate pair passing the required primality standard is found.

First-hit mode is significantly faster than attempting to prove that the pair is the closest possible pair to the midpoint.

This is the preferred mode for extremely large E .

1.8 Exact $d_{\min}$ Mode

A second mode is used when the objective is to determine the smallest central deviation producing a prime–prime pair.

In exact $d_{\min}$ mode, all admissible deviations below the successful deviation must be eliminated or tested sufficiently strongly.

This mode is considerably more expensive.

If a pair is found at deviation d , it cannot be declared the true $d_{\min}$ unless every admissible deviation smaller than d has been resolved.

For large integers, exact $d_{\min}$ determination may therefore be computationally impractical even when a Goldbach pair itself is easy to find.

The distinction between first-hit and exact $d_{\min}$ modes prevents a common source of overstatement in computational work.

1.9 Candidate States

For each tested pair, four basic states are recorded:

P/P: both members are prime or pass the required probable-prime threshold;

P/C: the first member is prime and the second is composite;

C/P: the first member is composite and the second is prime;

C/C: both are composite, or at least one side is already known composite before full testing of the other is required.

In peripheral mode, p is usually chosen from a list of known small or moderate primes. The practical state then mainly depends on q .

In centered mode, both sides may require independent testing.

Sequences of these states are useful diagnostic objects.

In particular, long sequences without P/P are interpreted as local Goldbach deserts.

1.10 Central and Peripheral Deserts

A central desert is an interval of deviations d in which no P/P pair is detected.

A peripheral desert is an interval of tested small or moderate primes p for which $q = E - p$ remains composite.

Two different quantities must be distinguished.

The first is an exhaustive desert length. This can be claimed only when the entire relevant interval has been checked without a P/P event.

The second is an observed cold-band extent. This refers to a sampled or partially tested region in which a long sequence of failures is observed.

A sampled cold band must never be reported as an exhaustive desert.

1.11 Admissibility Versus Synchronization

The experiments suggest a useful distinction between two sources of difficulty.

The first is low admissible density.

In this case, small-prime congruences eliminate most candidates before strong testing.

The second is synchronization failure.

Here, many candidates survive the sieve, but simultaneous primality of p and q occurs much later than expected.

To formalize this experimentally, let $A(E)$ denote a characteristic spacing between admissible candidates, and let $K(E)$ denote the number or rank of admissible candidates encountered before the first P/P event.

A useful heuristic decomposition is then $d_{\min}$ approximately $A(E) \times K(E)$.

Large values of $d_{\min}$ may therefore arise either because admissible candidates are sparse or because many admissible candidates fail before simultaneous primality occurs.

Our experiments indicate that some of the most difficult cases belong primarily to the second category.

1.12 Adaptive Search Zones

The algorithm uses three principal search regions.

The central region is centered around $m = E/2$.

The peripheral region uses small or moderate p and $q = E - p$.

An intermediate region may be explored near a square-root-type scale when neither the center nor the extreme periphery is productive.

The search does not assume that one region is universally optimal. Instead, the algorithm can switch between regions according to observed survivor density and primality outcomes.

A long C/C chain is interpreted as evidence that the current region may be locally cold.

The algorithm can then jump rather than continue a purely sequential scan.

1.13 The SURGERY Diagnostic Mode

When a case becomes unusually difficult, the algorithm stops ordinary progression and enters a diagnostic mode called SURGERY. The following quantities are examined: admissible density; spacing between survivors; largest survivor gaps; survival

rate as sieve depth increases; fraction of candidates eliminated on the p side; fraction eliminated on the q side; frequency of P/C and C/P events; local asymmetry; distance to neighboring primes when computationally feasible; persistence of cold bands over increasing scales.

The objective is not merely to continue searching but to determine why the current region is difficult.

The diagnostic can then indicate whether to deepen the sieve, jump to a different deviation range, switch search zone, or continue locally.

1.14 Stopping Condition and Final Audit

The search stops when a reliable P/P pair has been obtained according to the required standard.

Before reporting the pair, a strict final audit is performed.

The following conditions must all be checked: E is the intended even integer; p and q have the expected number of digits; $p + q = E$ exactly; p and q have the correct parity; the claimed primality status matches the actual test performed; probable-prime results are not reported as certified primes; external certification, when used, is recorded separately.

This final audit is particularly important for numbers containing thousands of decimal digits, where a single misplaced decimal exponent or additive constant can invalidate an otherwise correct computation.

The next section describes how these components are assembled into the optimized adaptive procedure used in the experiments.

2. Optimized Adaptive Algorithm

The algorithm used in this study is designed for isolated very large even integers, where exhaustive enumeration of all possible Goldbach decompositions is impractical and repeated primality testing of arbitrary candidates is computationally expensive.

The optimized procedure combines deep preliminary sieving, survivor ranking, adaptive search zone selection, selective probable-prime testing, and diagnostic analysis of difficult regions.

2.1 Input and Preliminary Audit

The input is an even integer E . The first step is to compute $m = E/2$. Before any search begins, a structural audit is performed.

The audit includes:

- parity of E ;
- exact decimal length of E ;
- decimal length of m ;
- residues of E modulo selected small primes;
- residues of m modulo selected small primes;
- small prime factors of E when easily detectable;
- local congruence restrictions on centered deviations;
- expected density of admissible candidates;

• identification of obvious residue classes that should be avoided. This preliminary audit is inexpensive and can immediately alter the order in which search regions are explored.

2.2 Choice of Initial Search Mode

The algorithm can start in one of three regions:

1. peripheral;
2. central;
3. intermediate or square-root-type region.

For a general first-hit search, the preferred practical strategy is often to begin in the peripheral region because p is then small and already known to be prime.

For investigations of logarithmic Goldbach windows or central deserts, the algorithm starts near $m = E/2$.

The intermediate region is principally an adaptive fallback when neither the center nor the extreme periphery is productive.

The present method does not assume that one zone is universally optimal.

2.3 Peripheral Generation

In peripheral mode, prime values p are generated in increasing order or according to a priority ranking. For every p , $q = E - p$.

The candidate is immediately rejected if q is divisible by a small sieve prime.

Because p is already prime, the computational problem reduces to identifying those p for which q survives increasingly deep sieving. The peripheral mode therefore has a major asymmetry advantage: only q normally requires a difficult primality test.

This becomes increasingly valuable as E grows.

2.4 Centered Generation

In centered mode, candidate deviations d is generated and $p = m - d$, $q = m + d$ is formed.

Before constructing or testing the full integers, modular conditions are applied to d . For every selected sieve prime r , deviations producing $m - d \equiv 0 \pmod{r}$ or $m + d \equiv 0 \pmod{r}$ are rejected.

Thus, the algorithm searches only through admissible deviations.

When the objective is to investigate a logarithmic window, candidate deviations are initially concentrated inside H_2 , H_3 , or H_∞ .

If no P/P event is found, the search can extend beyond the original window while preserving the observed desert information.

2.5 Max-Sieve First

After candidate generation, the deepest reasonably affordable sieve is applied.

This is the central optimization of the method. Instead of the conventional sequence small sieve $\rightarrow$ probable-prime test $\rightarrow$ repeat, the present method attempts deep sieve $\rightarrow$ rank survivors $\rightarrow$ probable-prime test only for premium candidates.

If B denotes the sieve bound, candidates are repeatedly filtered as B increases.

For example, one may use stages such as

$B = 100$,

$B = 1,000$,

$B = 10,000$,

$B = 100,000$, $B = 1,000,000$, depending on available computational resources.

The exact sequence is not fixed.

The practical objective is to make the final survivor population as small as possible before invoking costly strong primality tests.

2.6 Resistance Profile

For each candidate x , a resistance profile can be recorded.

If the candidate survives all sieve levels up to B , its resistance is at least B .

Candidates surviving only shallow levels are discarded.

Candidates surviving the deepest levels are designated premium survivors.

This concept provides a natural ranking even when no explicit probabilistic score is used.

A candidate surviving a much deeper sieve has already escaped a larger set of obvious compositeness mechanisms.

2.7 Selective Strong Testing

Only premium survivors are subjected to strong probable-prime tests.

The preferred sequence is:

1. deterministic modular filtering;
2. deep trial-divisor or wheel-based sieving;
3. Miller–Rabin testing with several bases;
4. Baillie–PSW-type probable-prime testing when available;
5. final certification when required.

The algorithm distinguishes carefully among: composite; survivor; probable prime; certified prime.

No probable-prime result is promoted automatically to certified-prime status.

2.8 Test the Cheaper Side First

When both p and q are large, the algorithm first tests the side that is computationally cheaper or has the highest expected probability of early elimination.

In many centered searches, this can save approximately half of the expensive operations.

If the first tested side is composite, the second side does not need to be tested.

The state is then recorded as C/unknown or reduced directly to a failed candidate depending on the purpose of the experiment.

When detailed chain statistics are required, both sides may be tested to distinguish C/C, C/P, and P/C.

2.9 Detection of Cold Regions

The search continuously records local outcome sequences.

A long run without P/P is considered evidence of a cold region. Examples include: C/C, C/C, C/C, C/C, ... or mixed sequences such as P/C, C/P, C/C, P/C, C/C, ... with no simultaneous P/P. The algorithm does not interpret such chains as evidence against Goldbach. They are computational signals indicating that continued sequential scanning may be inefficient.

2.10 Adaptive Jumps

When a cold region persists, the algorithm can jump. In centered mode, jump lengths can be related to H2 or H3. In peripheral mode, prime indices or ranges of p can be skipped when a region displays low productivity after sufficiently deep analysis. The algorithm can also change search zone: center $\rightarrow$ periphery, periphery $\rightarrow$ center, center $\rightarrow$ square-root region, or another adaptive sequence. The decision is empirical and based on survivor density, resistance, local gaps, and observed primality outcomes.

2.11 Surgery Mode

A difficult case triggers SURGERY mode. The following measurements are then collected:

- number of raw candidates;
- number of candidates surviving each sieve depth;
- survivor density;
- mean spacing between survivors;
- maximum survivor gap;
- first-prime-side success rate;
- second-prime-side success rate;
- P/C frequency;
- C/P frequency;
- C/C chain length;
- persistence of cold regions;
- feasible distances to neighboring primes.

The purpose of SURGERY is to determine whether the difficulty is caused mainly by the modular sieve geometry or by late-stage synchronization failure.

If admissible density is extremely low, the problem is primarily geometric.

If admissible density remains substantial but P/P never occurs, the region is interpreted as a synchronization desert.

2.12 Peripheral Priority Variants

In selected experiments, peripheral candidates can be ranked using additional arithmetic information.

For example, a score based on quantities such as $\gcd(p-1, E-2)$ and $\gcd(p+1, E+2)$ can be used as an empirical ranking feature. Such scores are not theoretical guarantees.

They are used only to prioritize candidate order.

One experiment in the present series identified a successful pair for $E = 10^{1725} + 554$ with $p = 73$, $q = 10^{1725} + 481$, after $p = 73$ appeared as a high-priority candidate under this type of ranking.

2.13 First-Hit Versus Exhaustive Mode

The optimized algorithm normally operates in first-hit mode.

It stops at the first reliable P/P candidate.

This is the appropriate mode when the computational objective is simply to construct one Goldbach representation.

Exact $d_{\min}$ determination requires exhaustive verification of all smaller admissible deviations.

Therefore, a pair found at deviation d is not labeled as the minimum central pair unless all smaller admissible deviations have been resolved.

This distinction is maintained throughout the reported experiments.

2.14 Final Validation Protocol

Every reported pair must satisfy $p + q = E$ exactly.

The final validation additionally checks:

- correct exponent in E ;
- correct additive offset;
- exact decimal length;
- correct signs in p and q ;
- parity;
- primality status;
- consistency between internal probable-prime results and external certification.

This final audit is mandatory because transcription errors become a serious risk when explicit integers contain thousands of digits.

2.15 Operational Pseudocode

The practical procedure can be summarized as follows.

Input: even integer E .

Step 1. Compute $m = E/2$.

Step 2. Audit residues and local arithmetic structure.

Step 3. Choose initial region.

Step 4. Generate admissible candidates.

Step 5. Apply the deepest feasible sieve.

Step 6. Rank surviving candidates by resistance.

Step 7. Test premium survivors only.

Step 8. Test the cheaper side first when both sides are large.

Step 9. Record candidate state.

Step 10. Stop immediately if a reliable P/P pair is found.

Step 11. If a long cold chain develops, activate SURGERY.

Step 12. Decide whether to deepen the sieve, jump, or change search region.

Step 13. Repeat until success or until the predefined computational budget is exhausted.

Step 14. Perform the final arithmetic and primality audit.

This algorithm is heuristic and computational. It does not by itself establish that the process must terminate for every even integer.

3. Experimental Results

The algorithm was tested on a large collection of very large even integers. The experiments were designed to include both easy and difficult cases, centered and peripheral searches, logarithmicwindow searches, and deliberately challenging central deserts.

The most important observations are summarized below.

3.1 Centered Probable-Prime Pairs

Several large centered pairs were obtained using deviations from $E/2$.

For

$E = 10^{1000} + 2$, a centered PRP/PRP pair was obtained with $d = 2,304,238$, $p = 5 \times 10^{999} - 2,304,237$, $q = 5 \times 10^{999} + 2,304,239$.

For

$E = 10^{1500} + 2$, a centered PRP/PRP pair was also obtained, with the first successful deviation recorded as $d = 5,749,478$.

For

$E = 10^{2000} + 2$, the corresponding centered experiment produced $d = 6,313,132$, with $p = 5 \times 10^{1999} - 6,313,131$, $q = 5 \times 10^{1999} + 6,313,133$.

These examples demonstrate that central search remains computationally practical for integers containing approximately 1000–2000 digits, although the cost increases substantially with size.

3.2 A Small-Deviation Centered Example Near 10^{2600}

A particularly favorable centered case was found for $E = 10^{2600} + 722$. Here $m = 5 \times 10^{2599} + 361$ and a successful centered deviation was $d = 1582$.

The corresponding pair is $p = 5 \times 10^{2599} - 1221$, $q = 5 \times 10^{2599} + 1943$.

Both members were confirmed as strong probable primes in the computational workflow used for this study.

The extremely small deviation relative to the size of E illustrates how strongly local difficulty can vary between even integers of comparable decimal length.

3.3 Certified Centered Pair at 10^{3000}

One of the strongest centered examples obtained in the project is $E = 10^{3000}$.

A centered pair was found at $d = 805797$. Thus $p = 5 \times 10^{2999} - 805797$, $q = 5 \times 10^{2999} + 805797$.

Both p and q were independently confirmed prime by the user during the verification phase.

This therefore constitutes a certified prime-prime Goldbach representation within the experimental record of this work.

The example is significant because it shows that centered search remains possible at the 3000-digit scale, although the general cost of probable-prime testing is already noticeably higher than in the

1000–2000-digit range.

3.4 Certified Centered Pair Near 10^{1000}

For

$E = 10^{1000} + 558$, the centered search produced $d = 2,069,958$.

The resulting p and q were subsequently confirmed prime by the user.

This case is included among the certified centered examples rather than the PRP-only examples.

3.5 Peripheral Representations

Peripheral search proved particularly effective in several large cases.

Instead of requiring two independent large primality tests near $E/2$, the method selects a small known prime p and tests only $q = E - p$. Several certified representations were obtained in this way.

For

$E = 10^{1750} + 444$, a successful pair is $p = 27191$, $q = 10^{1750} - 26747$.

The user confirmed q prime.

For

$E = 10^{2500} + 8$, a peripheral pair was obtained with $p = 25391$, $q = 10^{2500} - 25383$.

The user confirmed q prime using an external primality-verification service.

This example is particularly instructive because the same E had previously exhibited a very difficult centered search.

Thus, an even integer can simultaneously possess a long central desert and an easy peripheral Goldbach representation.

For

$E = 10^{2750} + 6528$, a certified peripheral representation was obtained with $p = 59159$, $q = 10^{2750} - 52631$.

20

For

$E = 10^{2767} + 266$, the pair $p = 83417$, $q = 10^{2767} - 83151$ was externally confirmed.

For

$E = 10^{3000} + 938$, a peripheral pair was found with $p = 4967$, $q = 10^{3000} - 4029$.

The user reported external confirmation of q as prime.

For

$E = 10^{1366} + 772$, a peripheral pair was obtained with $p = 40939$, $q = 10^{1366} - 40167$.

The user confirmed q prime using two independent external services.

3.6 Very Small Peripheral Prime

An especially striking peripheral example was obtained for $E = 10^{1725} + 554$. The pair is $p = 73$, $q = 10^{1725} + 481$.

The large q was externally confirmed prime.

This example demonstrates that the smallest successful peripheral p can occasionally be extremely small even when E contains more than one thousand digits.

The result also illustrates why a peripheral pass can be computationally worthwhile before committing to an expensive central search.

3.7 Record at 10^{6000}

The largest certified representation obtained in the present experimental series is $E = 10^{6000}$.

A peripheral pair was found with $p = 11213$,

$q = 10^{6000} - 11213$.

The user independently confirmed q as prime using FactorDB and

dCode.

Since $p = 11213$ is prime, this constitutes a certified prime-prime Goldbach representation according to the external verification reported during the project.

This 6000-digit case is the largest verified example currently included in the present algorithmic dataset.

It is important, however, not to interpret the existence of this rapid peripheral success as evidence that all 6000-digit even integers are equally easy.

The experiment demonstrates accessibility, not uniform complexity.

3.8 Additional Certified Peripheral Cases

Other certified or externally confirmed peripheral cases recorded during the project include: $E = 10^44000 + 44$, for which a prime-prime peripheral representation was confirmed by the user; $E = 10^43500 + 96$, for which both members of a peripheral representation were reported prime after external verification.

These cases extend the observed successful range between the 3000-digit centered example and the 6000-digit record.

3.9 Difficult Centered Cases Near 10^42500

The experiments also included deliberately difficult cases.

For

$E = 10^42500 + 6$, no centered double-PRP pair was verified up to a deviation exceeding approximately 10.6 million during the recorded search.

For

$E = 10^42500 + 8$, no centered double-PRP pair was found up to a deviation exceeding approximately 6.1 million in the corresponding difficult search.

These are experimental lower bounds on the scanned central region, not theoretical statements about $d_{\min}$.

The second case is especially informative because a peripheral certified pair was later found quickly.

This confirms that central difficulty and global Goldbach difficulty should not be conflated.

3.10 Comparison of an Easy and a Difficult 2500-Digit Case

A detailed sieve comparison was performed between an easy and a difficult case near 10^42500 .

$E = 10^42500 + 9248$, the centered search produced a successful pair at a very small deviation, $d = 1507$.

In the first 200,000 deviations, using sieve primes up to 1000, approximately 3,526 deviations survived.

This corresponds to about 3.526 percent of the scanned deviations.

The mean spacing between survivors was approximately 56.7, and the largest observed survivor gap was approximately 540.

For

$E = 10^42500 + 6$, the same sieve produced approximately 1,715 survivors in the first 200,000 deviations.

This corresponds to approximately 1.715 percent.

The mean spacing between survivors was approximately 116.7, and the largest survivor gap was approximately 912.

Thus, the difficult case had roughly half the admissible density of the easy case.

However, this factor is far too small to explain the difference between d approximately 1500 and a desert extending beyond ten million.

The principal difficulty therefore appears after small-prime sieving.

3.11 Synchronization Interpretation

The previous comparison motivates a decomposition of search difficulty into two components.

The first is admissibility geometry.

This is determined largely by congruence restrictions and is directly visible through sieve survivor density.

The second is prime synchronization.

A candidate can survive all small-prime restrictions while still failing because one or both components have larger prime divisors. In difficult cases, a substantial population of admissible deviations can exist without producing a P/P event for a long interval.

The experiments therefore support the terminology synchronization desert.

This concept is empirical and does not imply an underlying theorem.

3.12 Behavior with Increasing Decimal Length

The accumulated computational experiments suggest three practical regimes.

For E with up to approximately 1000 digits, both centered and peripheral searches are generally comfortable.

Between approximately 1000 and 2000 digits, the algorithm remains effective, especially after deep preliminary sieving.

Between approximately 2000 and 3000 digits, the cost of strong probable-prime testing becomes increasingly visible.

Beyond approximately 3000 digits, the sieve remains useful, but repeated primality testing of the surviving large integers increasingly dominates runtime.

This leads to a practical division:

high-efficiency regime:

up to approximately 10^42000 ; intermediate regime:

approximately 10^42000 to 10^43000 ; high-cost regime: beyond approximately 10^43000 .

These values should not be interpreted as intrinsic mathematical thresholds.

An easy 6000-digit peripheral case may be found faster than a difficult 2500-digit centered case.

Local arithmetic structure remains a major determinant of actual runtime.

3.13 Summary Table of Principal Examples

The principal experimental examples can be summarized as follows.

Even integer E Search mode Main parameter			Status
$10^{1000} + 2$	centered	$d = 2,304,238$	PRP/PRP
$10^{1000} + 558$	centered	$d = 2,069,958$	prime-prime confirmed
$10^{1500} + 2$	centered	$d = 5,749,478$	PRP/PRP
$10^{1725} + 554$	peripheral	$p = 73$	prime-prime confirmed
$10^{1750} + 444$	peripheral	$p = 27191$	prime-prime confirmed
$10^{2000} + 2$	centered	$d = 6,313,132$	PRP/PRP
$10^{2500} + 8$	peripheral	$p = 25391$	prime-prime confirmed
$10^{2500} + 9248$ centered	$d = 1507$	PRP/PRP	
$10^{2600} + 722$	centered	$d = 1582$	PRP/PRP
$10^{2750} + 6528$ peripheral	$p = 59159$	prime-prime confirmed	
$10^{2767} + 266$	peripheral	$p = 83417$	prime-prime confirmed
10^{3000}	centered	$d = 805797$	prime-prime confirmed
$10^{3000} + 938$	peripheral	$p = 4967$	prime-prime confirmed
$10^{3500} + 96$	peripheral	recorded pair	prime-prime confirmed
$10^{4000} + 44$	peripheral	recorded pair	prime-prime confirmed
10^{6000}	peripheral	$p = 11213$	prime-prime confirmed

A larger catalogue, including additional PRP pairs, negative scans, difficult cases, and verification notes, is provided in Appendix 1.

3.14 Main Experimental Conclusions

The experiments support five principal computational conclusions. First, deep modular sieving remains useful even for integers containing several thousand digits.

Second, the expensive part of the calculation progressively shifts from sieving to primality testing as E increases.

Third, central and peripheral difficulty can differ dramatically for the same E.

Fourth, low admissible density alone does not explain the longest observed Goldbach deserts.

Fifth, a flexible multi-zone algorithm is substantially more practical than a rigid search confined permanently to the center or to the periphery.

These observations motivate the performance analysis and comparative discussion developed in the following sections.

4. Performance and Practical Limits

The experiments reported in this study reveal a clear change in computational behavior as the decimal length of E increases.

This change does not correspond to a mathematical boundary of the Goldbach problem. It is a practical transition caused mainly by the increasing cost of strong primality testing on very large integers.

4.1 High-Efficiency Regime

For even integers up to approximately the 10^{1000} scale, the

algorithm is generally comfortable to operate.

Deep modular sieving eliminates a large fraction of candidates quickly, and the remaining probable prime tests are still sufficiently inexpensive to allow repeated exploration of both central and peripheral regions.

The same favorable behavior persists, although with increasing cost, through much of the interval up to approximately 10^{2000} .

In this range, the MAX-SIEVE FIRST strategy is especially effective because a large number of candidates can be discarded before strong testing.

The algorithm therefore spends most of its time only on a relatively small set of premium survivors.

4.2 Intermediate Regime

Between approximately 10^{2000} and 10^{3000} , the balance begins to change.

The sieve remains effective.

Candidate generation, residue filtering, wheel-based elimination, and survivor ranking are still computationally manageable.

However, the cost of testing a surviving q increases substantially because q now contains several thousand decimal digits.

Repeated Miller–Rabin or Baillie–PSW-type tests therefore begin to dominate the search.

The practical importance of candidate ranking increases in this regime.

A weakly filtered search can waste substantial time testing candidates that would have been eliminated cheaply by a deeper sieve.

4.3 High-Cost Regime Beyond Approximately 10^33000

Beyond approximately 10^33000 , the main bottleneck is no longer the generation of candidates. The bottleneck becomes the transition premium survivor $\rightarrow$ strong primality test.

This was clearly observed in several recent experiments near 10^33900 .

Deep sieving can still isolate a manageable list of highly resistant peripheral candidates.

However, confirming each associated q by a strong probable-prime test becomes noticeably more expensive.

Thus, the algorithm does not cease to function.

Rather, the marginal cost of each final candidate increases.

This distinction is important.

The sieve remains scalable much farther than the strong-testing stage.

4.4 Why There is no Sharp Size Limit

The practical threshold near 10^33000 should not be interpreted as a hard limit.

Search difficulty depends strongly on the local arithmetic structure of E .

A favorable peripheral case can be solved rapidly even when E contains far more than 3000 digits.

The clearest example in the present dataset is $E = 10^6000$, for which the peripheral pair $p = 11213$, $q = 10^6000 - 11213$ was obtained and externally confirmed.

By contrast, some 2500-digit centered cases produced long deserts extending over millions of deviations.

Therefore, decimal length is only one component of complexity.

Local congruence structure and simultaneous prime synchronization can be equally important.

4.5 Central Versus Peripheral Computational Cost

Centered search is intrinsically more expensive when both p and q are very large. In the centered representation, $p = E/2 - d$, $q = E/2 + d$, both numbers generally require independent strong testing.

If the first tested side is composite, testing the second side can be skipped.

Nevertheless, the average cost remains higher than in peripheral mode.

In peripheral mode, p is usually selected from a list of known small or moderate primes.

Therefore, only q normally requires a large-integer primality test.

This asymmetry explains why peripheral search often remains practical at sizes where centered search becomes much slower.

4.6 Effect of Difficult Synchronization Deserts

The longest computational delays are not always caused by sparse admissible candidates.

In some hard cases, deep sieving continues to produce many survivors, but none of them yields simultaneous primality.

This forces the algorithm to perform many expensive strong tests before reaching a P/P event.

The total runtime can therefore increase dramatically even when

the sieve itself behaves normally.

This is why the distinction between admissibility geometry and synchronization difficulty is operationally important.

A low-density case can sometimes be cheap because very few candidates require final testing.

A high-density synchronization desert can be more expensive because many premium survivors must be tested before success.

4.7 Practical Operating Regimes

Based on the present experimental campaign, the following empirical classification is useful.

High-Efficiency Regime

E up to approximately 10^42000 .

Typical behavior:

- deep sieve is highly effective;
- probable-prime testing remains practical;
- both centered and peripheral modes can be explored repeatedly;
- exact or near-exact central searches may still be feasible in favorable cases.

Intermediate Regime

Approximately $10^42000 < E \leq 10^43000$.

Typical behavior:

- sieving remains efficient;
- strong-testing cost becomes increasingly important;
- survivor ranking becomes essential;
- first-hit mode becomes preferable to exact $d_{\min}$ mode.

High-Cost Regime

$E >$ approximately 10^43000 .

Typical behavior:

- candidate generation remains manageable;
- deep sieving still provides major savings;
- repeated strong primality testing dominates runtime;
- peripheral mode becomes increasingly attractive;
- exact centered scans can become prohibitively expensive in difficult cases.

These thresholds are empirical observations from the current implementation.

They are not universal complexity bounds.

4.8 Main Limiting Factor

The main practical limitation of the present algorithm can therefore be expressed as follows:

the method remains capable of finding strong candidates at very large scales, but the cost of confirming those candidates grows faster than the cost of generating them. In simplified form, candidate generation cost $\ll$ strong primality-testing cost for sufficiently large E .

This suggests that future algorithmic improvements should focus less on producing even more raw candidates and more on reducing the number of final large-integer tests.

Possible improvements include:

- deeper vectorized sieving;
- stronger wheel construction;
- improved survivor scoring;
- better ordering of candidate tests;
- parallel strong testing;
- adaptive switching to peripheral search;
- early elimination using additional arithmetic filters.

4.9 Practical Conclusion

The current implementation performs very well up to approximately the 10^{2000} scale.

It remains effective but progressively slower between approximately 10^{2000} and 10^{3000} .

Beyond approximately 10^{3000} , the principal computational difficulty is the strong testing of several-thousand-digit survivors. The algorithm should therefore not be described as failing beyond this region.

A more accurate statement is that its limiting stage changes.

The sieve continues to work efficiently, while primality confirmation becomes the dominant computational cost.

This observation is central to the optimization strategy proposed in this paper and motivates the discussion of synchronization deserts in the next section.

5. Goldbach Deserts and Synchronization

One of the most important observations of the present study is that difficult Goldbach searches cannot be explained solely by the density of candidates surviving small-prime sieving.

The experiments repeatedly showed that two even integers of similar size can display very different computational behavior even when their admissible survivor densities differ only moderately.

This motivates a distinction between two mechanisms: admissibility geometry, and prime synchronization.

5.1 Admissibility Geometry

For a centered search, $p = E/2 - d$, $q = E/2 + d$.

For each small prime r , deviations satisfying $p \equiv 0 \pmod r$ or $q \equiv 0 \pmod r$ are removed.

After sieving by all selected small primes, a deterministic set of admissible deviations remains.

The geometry of this admissible set can be described using several quantities:

- total number of survivors;
- survivor density;
- mean spacing;
- largest gap;
- clustering;
- local depletion;
- resistance under increasing sieve depth.

These quantities depend strongly on the residue structure of E .

They describe how much of the search space is removed before strong primality testing begins.

5.2 Why Admissibility Alone is Insufficient

A naive interpretation would be that the most difficult Goldbach

cases are simply those with the fewest admissible deviations.

The experiments do not support such a simple explanation.

The comparison between $E = 10^{2500} + 9248$ and $E = 10^{2500} + 6$ is particularly informative.

Using sieve primes up to 1000 over the first 200,000 deviations, the easy case retained approximately 3.526% of deviations, whereas the difficult case retained approximately 1.715%.

Thus, the difficult case had only about half the local survivor density. However, the successful deviation in the easy case was approximately $d = 1507$, while the difficult case showed no verified P/P event throughout a central scan extending beyond ten million deviations.

A factor of approximately two in admissible density cannot by itself explain a difference of several thousand in observed search depth.

Therefore, much of the difficulty must occur after the small-prime sieve.

5.3 Prime Synchronization

A surviving deviation only means that neither p nor q has been eliminated by the tested small primes.

It does not imply primality. The pair may still fail because:

p has a larger prime factor; q has a larger prime factor; or both are composite.

The event required by Goldbach is therefore not merely p admissible and q admissible, but p prime and q prime simultaneously.

This simultaneous event is referred to here as prime synchronization.

The term is computational rather than probabilistic in a strict theorem-level sense.

It is used to describe the final coincidence required after modular admissibility has already been satisfied.

5.4 Synchronization Deserts

A synchronization desert is an experimentally observed region in which many admissible candidates survive deep sieving, but no P/P event is detected for a long interval.

Such a region differs conceptually from a sieve desert.

A sieve desert contains very few admissible candidates.

A synchronization desert may contain many admissible candidates, but those candidates repeatedly fail at the primality stage.

This distinction is important for algorithm design.

If a region is sparse because of modular restrictions, deeper congruence analysis may explain most of the difficulty.

If the region remains dense in survivors but fails repeatedly at strong testing, the search problem is fundamentally different.

In that case, adding a slightly deeper small-prime sieve may produce only limited improvement.

5.5 $A(E)$ and $K(E)$

To separate these two effects, it is useful to introduce two empirical quantities.

Let $A(E)$ denote a characteristic spacing between admissible deviations in the searched region.

Let $K(E)$ denote the rank of the first successful P/P event among admissible survivors. Then a rough heuristic relation is $d_{\min}$ approximately $A(E) \times K(E)$.

This expression is not an identity.

Its purpose is to decompose a large observed deviation into two conceptually different contributions.

A large $A(E)$ means survivors are intrinsically sparse.

A large $K(E)$ means many admissible survivors must be tested before simultaneous primality appears.

The experiments suggest that some of the hardest cases are characterized primarily by unusually large $K(E)$, not by exceptionally large $A(E)$.

5.6 Chain Structure

The search process can also be represented by chains of local states. In a centered search, the basic states are

C/C ,

C/P ,

P/C , P/P .

A long sequence of C/C , C/C , C/C , ...

indicates a cold region. Mixed sequences such as C/P , P/C , C/C , P/C , C/P contain more information.

They show that primality appears separately on both sides, but simultaneous primality has not yet occurred.

Such mixed chains are particularly interesting because they suggest that the two prime processes are individually active but poorly synchronized.

A P/P event terminates the chain.

5.7 Exhaustive Deserts Versus Sampled Cold Bands

A strict distinction is maintained between an exhaustive desert and a sampled cold band.

An exhaustive desert can be claimed only when every admissible candidate in a continuous interval has been resolved and no P/P event occurs.

A sampled cold band is weaker.

It refers to a region in which many tested candidates fail, but where not every admissible candidate has necessarily been resolved.

This distinction is essential when thousands-of-digit primality tests become too expensive for exhaustive scanning.

The present work reports both types of observations but does not merge them.

5.8 Central Difficulty Versus Global Difficulty

One of the clearest experimental results is that a long central desert does not imply that the corresponding even integer is globally difficult.

The example

$$E = 10^{2500} + 8$$

illustrates this directly.

The centered search showed no verified P/P event through a very large deviation range. Yet a peripheral representation was later obtained with $p = 25391$, $q = 10^{2500} - 25383$, and the large q was externally confirmed prime.

Thus, the central region can be locally cold while the periphery is easy.

This observation strongly supports the adaptive multi-zone design of the algorithm.

5.9 Local Arithmetic Signatures

Different values of E produce different residue structures modulo small primes.

These structures influence:

- how many deviations survive;
- which deviations survive;
- clustering of survivors;
- local spacing;
- the relative behavior of the p and q sides.

The experiments suggest that each E may possess a local arithmetic signature.

This signature does not determine the exact first Goldbach pair, but it appears to influence the geometry of the search.

The difficult-versus-easy contrast near 10^{2500} is one example.

The same decimal scale can contain both very short and very long central searches.

Therefore, complexity is not controlled by size alone.

5.10 Effect on the Adaptive Algorithm

The distinction between admissibility and synchronization leads directly to algorithmic decisions.

If survivor density is extremely low:

deepen modular analysis; optimize the wheel; exploit residue structure; search for alternative admissible lanes.

If survivor density is reasonable but strong tests repeatedly fail: do not assume that a slightly deeper sieve will solve the problem; measure the chain structure; activate SURGERY; consider a jump; consider changing region; prioritize alternative premium survivors. This is one of the main reasons why the present algorithm is adaptive rather than purely sequential.

5.11 Surgery as a Desert Classifier

The SURGERY mode is especially useful for separating the two desert mechanisms.

For a difficult interval, the procedure records:

$N0$ = number of raw candidates; NB = number surviving sieve depth B ; survival fraction $NB/N0$; mean survivor spacing; largest survivor gap; number of strong tests; number of P/C events; number of C/P events; number of C/C events; rank of the first P/P event, if one occurs.

These measurements allow a difficult interval to be classified approximately as:

admissibility-limited; synchronization-limited; mixed.

This classification is empirical but operationally useful.

5.12 Why this Matters for Optimization?

The distinction changes where computational effort should be spent.

In an admissibility-limited case, a better sieve can provide a substantial benefit.

In a synchronization-limited case, the main cost is the number of strong primality tests required after sieving.

The best optimization is then not necessarily a deeper sieve.

It may instead be: better survivor ranking; better zone selection; larger jumps; parallel testing; or faster primality screening.

Thus, the observed desert mechanism determines the appropriate optimization strategy.

5.13 Relation to the Logarithmic Windows

The logarithmic scales H_2 , H_3 , and H_∞ remain useful in this context.

They provide natural reference lengths for measuring:

- where the first admissible survivors occur;
- how long a central cold region persists;
- whether a P/P event appears inside a critical or supercritical window;
- how the observed $d_{\min}$ compares across different E .

However, the experiments show that the mere existence of many admissible deviations inside H_3 does not guarantee an immediate P/P event.

The key issue is still the transition from admissibility to simultaneous primality.

This is why the logarithmic windows are retained as heuristic and algorithmic scales rather than promoted to universal theorems.

5.14 Broader Computational Interpretation

The Practical Goldbach Search Can Therefore Be Viewed as a Two-Stage Process.

Stage 1:

construct a deterministic admissible skeleton using modular arithmetic.

Stage 2:

search that skeleton for simultaneous primality.

The first stage is comparatively well controlled.

The second stage is the source of most extreme variability.

This two-stage picture explains several observations that initially appeared contradictory: why a deep sieve can work well while the overall search remains slow; why two integers with similar survivor density can have very different first-hit distances; why a central desert can coexist with an easy peripheral pair; why very large integers can sometimes be easier than much smaller ones.

5.15 Main Conclusion of the Desert Analysis

The experimental evidence supports the following computational conclusion:

Goldbach search difficulty should not be modeled by candidate density alone. A more useful framework separates admissibility geometry from prime synchronization.

The longest observed difficult cases appear to be largely synchronization deserts rather than simple shortages of admissible candidates.

This interpretation does not constitute a theorem about the distribution of Goldbach pairs.

It is an empirical structural model derived from the behavior of the adaptive algorithm.

Its main value is practical: it indicates when the algorithm should deepen the sieve, when it should change search region, and when further local scanning is likely to waste expensive primality tests.

6. Discussion

The present work was designed as a computational study rather than as a proof of Goldbach's conjecture. Its main contribution is the organization of large-scale constructive search around a small number of operational principles: deep preliminary sieving, adaptive search-zone selection, survivor ranking, selective strong primality testing, and explicit diagnosis of difficult regions.

The experiments show that these principles remain useful even when E contains several thousand decimal digits. At the same time, they reveal that the computational difficulty of Goldbach search is highly non-uniform and cannot be predicted from decimal length alone.

6.1 Relation to Classical Goldbach Theory

The classical analytic theory of Goldbach's problem is dominated by methods that estimate the number of representations rather than construct one representation of one isolated enormous even integer.

The Hardy–Littlewood circle method predicts an asymptotic density of Goldbach representations and provides the natural probabilistic background for expecting many representations of large even integers [1,2].

Vinogradov's work on sums of three primes and Helfgott's later completion of the ternary Goldbach problem demonstrate the power of analytic methods in additive prime problems [3–5].

For the binary conjecture, however, the parity barrier remains decisive. Chen's theorem shows that every sufficiently large even integer can be represented as a prime plus a number having at most two prime factors, but it does not force the second summand to be prime [6].

The present algorithm does not attempt to replace these theories.

Instead, it addresses a different question:

given one specific very large even integer E , how can one identify a prime-prime representation with as few expensive tests as possible?

These constructive objective places the present work closer to computational number theory than to classical asymptotic Goldbach analysis.

6.2 Relation to Exhaustive Computational Verification

Large-scale computational verification of Goldbach has already established the conjecture throughout extremely large finite intervals [10].

Such verification is conceptually different from the present problem.

An exhaustive interval computation can exploit:

- segmented sieves;
- ordered traversal;
- shared prime tables;
- reusable modular information;
- interval-based memory organization.

In contrast, the present algorithm often receives one isolated integer such as 10^{3000} or 10^{6000} .

There is no nearby interval of manageable width whose full prime

structure can be precomputed.

The problem is therefore one of selective search.

The algorithm attempts to minimize the number of large-integer probable-prime tests by extracting only a small set of highly resistant candidates.

6.3 Importance of Max-Sieve First

The experiments strongly support the practical rule that sieving should be pushed as deeply as is computationally reasonable before strong probable-prime testing begins.

This principle is unsurprising from a general computational perspective but becomes particularly important for thousand-digit Goldbach searches.

For a small prime r , testing divisibility or residue admissibility is extremely cheap.

By contrast, modular exponentiation involving a 3000- or 6000-digit candidate is expensive.

Consequently, a shallow sieve followed by repeated Miller–Rabin tests is inefficient at extreme scale.

The present results indicate that the value of the sieve increases, rather than decreases, as E grows.

The limiting step shifts toward the surviving primality tests.

This observation is consistent with general principles of computational primality testing and fast arithmetic [11–15].

6.4 Probable Primes and Certified Primes

A central methodological requirement of the study is the strict separation between probable-prime status and proven primality.

Miller–Rabin testing provides an efficient probabilistic filter [11,12].

Baillie–PSW-type procedures provide an additional strong practical screen. However, neither should be described as a mathematical proof of primality.

For selected final candidates, external certification or dedicated primality-proving methods such as ECPP are more appropriate [13].

This distinction becomes increasingly important when results involve integers containing thousands of digits.

A computational paper should record not only the candidate pair but also the level of verification supporting the claim.

For this reason, the present work distinguishes explicitly among: survivor;

PRP;

externally confirmed prime; certified prime.

6.5 Central Windows and Heuristic Scales

The centered formulation $p = E/2 - d$, $q = E/2 + d$ provides a natural way to investigate whether Goldbach pairs occur close to the midpoint. The scales

$H_2 = (\log E)^2$, $H_3 = (\log E)^3$, and

$H_\infty = (\log E)^3 \log \log E$ were used throughout the project as heuristic and diagnostic windows.

The experimental evidence suggests that H_3 is substantially more robust than H_2 in practical searches.

However, the difficult cases near 10^{2500} show why these scales must not be presented as proven universal bounds.

A long central desert can exceed a chosen logarithmic window even while a simple peripheral representation exists.

Therefore, the present work treats logarithmic windows as algorithmic guides, not as theorems.

6.6 Why the Peripheral Strategy Matters

One of the strongest empirical conclusions is that peripheral search should not be viewed as a secondary or inferior version of centered search.

For large E , peripheral mode has a substantial computational advantage.

The small component p can be selected from an already known prime list, leaving only $q = E - p$ to be tested strongly. The record case $E = 10^{6000}$, $p = 11213$, $q = 10^{6000} - 11213$ illustrates this advantage. Likewise, the very small prime $p = 73$

in the representation of $E = 10^{1725} + 554$ shows that an enormous even integer can occasionally possess a representation that is extremely cheap to discover if the periphery is examined early.

Thus, a purely central algorithm would discard useful structure.

6.7 Central Deserts Do Not Imply Global Difficulty

The case

$E = 10^{2500} + 8$ is one of the most informative examples in the dataset.

The centered search showed a very long difficult region. Nevertheless, a peripheral pair was later found with $p = 25391$.

This single example has an important algorithmic consequence.

Difficulty is strongly local.

An even integer can be difficult in one geometric representation and easy in another.

This supports the adaptive architecture of the algorithm and argues against rigid search strategies.

6.8 Admissibility and the Parity Barrier

The modular sieve can describe admissibility very effectively.

For each small prime r , forbidden residue classes can be removed exactly.

As the sieve depth increases, the set of surviving candidates becomes increasingly resistant to local divisibility obstructions.

However, the sieve cannot force simultaneous primality.

This is closely related to the classical parity problem in sieve theory [6–8].

The experiments provide a computational manifestation of this limitation.

Even after small-prime obstructions are almost completely removed, long sequences of composite outcomes can persist.

The algorithm therefore reaches a point where additional local congruence information provides diminishing returns.

The remaining problem is genuinely about primality synchronization.

6.9 Synchronization Deserts as an Empirical Concept

The term synchronization desert was introduced to describe regions where many admissible candidates survive but no P/P event appears for a long time.

This concept should be interpreted carefully.

It is not proposed as a theorem or as a new asymptotic law.

Its value is diagnostic.

It explains why survivor density and total runtime can become decoupled.

A region may have a healthy survivor population but still require many expensive strong tests.

This is precisely the situation in which deeper sieving alone is unlikely to transform the computation dramatically.

Instead, the algorithm should improve candidate ranking or change search region.

6.10 Comparison with Conventional Sequential Search

A simple Goldbach search could proceed as follows: take primes p in increasing order; compute $q = E - p$; test q ; repeat.

This approach has the advantage of simplicity.

Its weakness is that it applies an expensive operation repeatedly without first extracting as much modular information as possible.

The present algorithm changes the order:

generate many candidates cheaply; sieve them deeply; retain a premium subset; test only the survivors.

This is conceptually similar to standard practice in computational prime searches, where inexpensive filters are used before expensive primality tests [15].

The added contribution here is the integration of that principle with centered logarithmic windows, adaptive zone changes, desert diagnostics, and Goldbach-specific bilateral admissibility.

6.11 Scaling Beyond 3000 Digits

The observed slowdown around 10^3000 is not an indication that the algorithm reaches a mathematical limit.

Instead, the strong-testing stage becomes the dominant cost.

This suggests several future computational improvements.

Parallelization is an obvious possibility because independent premium survivors can often be tested independently.

Batch modular exponentiation may also reduce overhead.

A stronger ranking function could reduce the number of candidates reaching the final testing stage.

Improved wheel construction could further compress the survivor set.

Finally, hybrid strategies could use different sieve depths depending on the expected cost of the final candidate.

These improvements are engineering questions rather than new mathematical principles.

6.12 The Role of Local Arithmetic Structure

The experiments strongly suggest that two integers with comparable size can have very different search profiles.

This difference is visible in: residue structure; survivor density; survivor gaps; frequency of one-sided prime events; length of C/C chains; location of the first P/P event.

The algorithm therefore treats the local arithmetic structure of E as a meaningful computational signature.

At present, no formula is known that predicts the first successful pair from this signature.

Nevertheless, the signature can guide search decisions.

This may become one of the most useful directions for future

optimization.

6.13 Limits of the Present Study

Several limitations should be stated explicitly.

First, the numerical experiments are not exhaustive over all even integers of a given digit length.

Second, many very large examples were screened using probable-prime tests before external confirmation.

Third, runtime comparisons were obtained across an evolving experimental workflow rather than a single fixed benchmark platform.

Fourth, the empirical thresholds around 10^2000 and 10^3000 depend on implementation and hardware.

Fifth, the synchronization-desert interpretation remains empirical.

Sixth, no universal upper bound for $d_{\min}$ or for the first successful peripheral p is proved.

Finally, the algorithm itself does not imply Goldbach's conjecture.

These limitations do not weaken the computational observations, but they define their proper scope.

6.14 Relation to a Possible Proof Program

Although the present article is algorithmic, the experiments have theoretical relevance.

They reinforce the distinction between local admissibility and final parity-sensitive primality correlation.

The associated proof program developed in parallel reduces the remaining obstruction to a uniform cancellation problem involving Möbius-weighted prime correlations.

That reduction is described separately in Appendix 2.

The computational results do not prove the required cancellation.

However, the observed persistence of admissible candidates through deep sieving is consistent with the idea that the final obstruction lies after local congruence elimination rather than inside it.

This is a conceptual link between the algorithmic and analytic sides of the project.

6.15 Reproducibility

Reproducibility is especially important for computations at extreme decimal scale.

For each final pair, the following information should be preserved: the exact value of E ; the search region; the value of p ; the value of q or its compact exact expression; the deviation d when centered; the sieve depth; the probable-prime method; the certification method; the exact identity $p + q = E$.

Appendix 1 provides the catalogue of principal examples.

Appendix 3 gives the recommended operational procedure for reproducing and extending the algorithm.

6.16 Comparison with the Broader Literature

The present work lies at the intersection of several established areas.

The theoretical background comes from additive prime theory and the circle method [1–5].

The limitations of purely sieve-based detection are connected with the parity phenomenon and classical sieve theory [6–8].

The almost-all perspective and exceptional-set analysis belong to the analytic study of binary Goldbach [9].

The finite verification perspective is represented by large-scale Goldbach computations [10].

The final numerical stage depends on modern primality testing and certification [11–15].

The contribution of the present study is therefore not a replacement for any of these areas.

It is a Goldbach-specific search architecture for isolated extremely large even integers.

6.17 Main Interpretation

The experiments support a simple but useful picture.

The Goldbach search problem for a fixed huge E contains two computational layers.

The first is deterministic and local:

remove candidates forbidden by congruences.

The second is sparse and expensive:

find a surviving candidate whose two components are simultaneously prime.

The first layer scales surprisingly well.

The second becomes increasingly expensive with digit length and can display severe local fluctuations.

This separation explains why the algorithm remains structurally effective even when runtime increases sharply.

It also explains why future gains are likely to come from reducing the number of final strong tests rather than from generating still more raw candidates.

6.18 Discussion Conclusion

The present results indicate that adaptive Goldbach pair search remains feasible far beyond the range in which naive repeated primality testing would be attractive.

The principal computational gains arise from combining deep sieving with flexible search geometry.

At the same time, the experiments expose a persistent synchronization problem that cannot be removed by local modular filtering alone.

This observation defines both the current practical limit of the algorithm and the main direction for future optimization.

The method should therefore be viewed as a scalable constructive framework rather than as a fixed search formula.

Its strongest feature is not that one particular zone always succeeds, but that the algorithm can recognize when a zone is becoming inefficient and reorganize the search before the most expensive stage is repeated unnecessarily.

7. Conclusion

This study presents an adaptive computational framework for constructing Goldbach prime pairs in isolated very large even integers.

The method combines centered and peripheral search, logarithmic search scales, deep modular sieving, survivor ranking, selective strong primality testing, adaptive zone switching, and a diagnostic analysis of difficult regions.

The central computational principle is MAX-SIEVE FIRST.

Rather than applying expensive probable-prime tests to large numbers of weak candidates, the algorithm attempts to eliminate as many candidates as possible through inexpensive modular arithmetic before strong testing begins.

The experiments show that this strategy remains useful even when E contains several thousand decimal digits.

The computational campaign produced a substantial collection of centered and peripheral Goldbach representations.

Among the most important examples are confirmed or probable-prime pairs near

10^{1000} ,

10^{1500} ,

10^{1750} ,

10^{2000} ,

10^{2500} ,

10^{2600} ,

10^{2750} ,

10^{2767} ,

10^{3000} ,

10^{3500} , 10^{4000} , and

10^{6000} .

The largest confirmed example in the present dataset is $E = 10^{6000}$, with $p = 11213$ and $q = 10^{6000} - 11213$.

This result demonstrates that very large Goldbach representations can remain computationally accessible when the local arithmetic structure is favorable.

The experiments also reveal that decimal length alone does not determine search difficulty.

Some 2500-digit centered cases produced deserts extending over millions of deviations, whereas much larger peripheral cases were solved rapidly.

This difference led to one of the main structural observations of the study: Goldbach search difficulty should be separated into admissibility geometry and prime synchronization.

The first mechanism concerns the density and arrangement of candidates surviving small-prime congruence restrictions.

The second concerns the eventual occurrence of simultaneous primality among those survivors.

The longest observed difficult cases appear to be dominated primarily by synchronization failure rather than by a simple shortage of admissible candidates.

This distinction has direct algorithmic consequences.

When admissible density is low, deeper sieving and improved modular analysis are appropriate.

When many resistant survivors remain but no P/P event occurs, the algorithm should instead improve candidate ordering, change search region, increase jump length, or parallelize expensive strong tests.

The study also identifies a practical transition in computational cost.

The present workflow remains highly effective up to approximately the 10^{2000} scale.

Between approximately 10^{2000} and 10^{3000} , strong primality testing becomes increasingly important.

Beyond approximately 10^3000 , the sieve itself can remain efficient while the repeated testing of several-thousand-digit survivors becomes the dominant computational bottleneck.

These thresholds are empirical operating regimes and should not be interpreted as mathematical limits.

A principal advantage of the adaptive method is that it does not assume that the best representation must occur in a fixed region.

A long central desert can coexist with an easy peripheral representation, and a difficult peripheral interval may justify returning to the center or another search scale.

The algorithm therefore treats search geometry dynamically rather than rigidly.

The present work does not prove Goldbach's conjecture.

In particular, no universal logarithmic bound for the first centered Goldbach pair is established, and no universal bound is proved for the first successful peripheral prime.

The logarithmic scales

$H_2 = (\log E)^2$, $H_3 = (\log E)^3$, and

$H_\infty = (\log E)^3 \log \log E$ are used only as heuristic and computational reference scales.

The theoretical reduction developed in parallel with this work is presented separately in Appendix 2.

Its final uniform Möbius-cancellation condition remains unproved and is therefore not used as a theorem in the present article.

The main contribution of this study is consequently computational and structural.

It provides:

- a reproducible search architecture;
- a deep-sieve-first strategy;
- a distinction between first-hit and exact-minimum modes;
- an adaptive multi-zone procedure;
- a diagnostic framework for Goldbach deserts;
- a catalogue of very large explicit examples;
- an empirical analysis of the practical limits of the method.

Future improvements should concentrate primarily on reducing the cost of the final strong-testing stage.

The most promising directions include deeper vectorized sieving, parallel testing of premium survivors, improved survivor scoring, adaptive batching, and more effective prediction of locally favorable search zones.

The current experiments suggest that the next major algorithmic improvement will probably not come from generating more candidates.

It will come from identifying which survivors are worth testing.

In this sense, the present work transforms the computational Goldbach problem from a broad enumeration problem into a selective search problem:

construct a thin admissible skeleton, rank its most resistant

members, and concentrate the expensive primality effort only where simultaneous primality is most plausible.

This adaptive framework does not resolve Goldbach's conjecture, but it provides a practical route for constructing and studying Goldbach pairs at numerical scales far beyond ordinary direct search.

References

1. Hardy, G. H., and Littlewood, J. E. (1923). Some problems of Partitio Numerorum. III. On the expression of a number as a sum of primes. *Acta Mathematica*, 44, 1–70.
2. Hardy, G. H., and Littlewood, J. E. (1924). Some problems of Partitio Numerorum. V. A further contribution to the study of Goldbach's problem. *Proceedings of the London Mathematical Society*, 22, 46–56.
3. Vinogradov, I. M. (1937). Representation of an odd number as a sum of three primes. *Doklady Akademii Nauk SSSR*, 15, 291–294.
4. Helfgott, H. A. (2013). The ternary Goldbach conjecture is true.
5. Helfgott, H. A. (2014). The ternary Goldbach problem. 2014–2015.
6. Chen, J. R. (1973). On the representation of a large even integer as the sum of a prime and the product of at most two primes. *Scientia Sinica*, 16, 157–176.
7. Halberstam, H., and Richert, H.-E. (1974). *Sieve Methods*. Academic Press, London.
8. Iwaniec, H., and Kowalski, E. (2004). *Analytic Number Theory*. American Mathematical Society, Providence.
9. Montgomery, H. L., and Vaughan, R. C. (1975). The exceptional set of Goldbach's problem. *Acta Arithmetica*, 27, 353–370.
10. Oliveira e Silva, T., Herzog, S., and Pardi, S. (2014). Empirical verification of the even Goldbach conjecture and computation of prime gaps up to 4×10^{18} . *Mathematics of Computation*, 83, 2033–2060.
11. Miller, G. L. (1976). Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13, 300–317.
12. Rabin, M. O. (1980). Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12, 128–138.
13. Atkin, A. O. L., and Morain, F. (1993). Elliptic curves and primality proving. *Mathematics of Computation*, 61, 29–68.
14. Agrawal, M., Kayal, N., and Saxena, N. (2004). PRIMES is in P. *Annals of Mathematics*, 160, 781–793.
15. Crandall, R., and Pomerance, C. (2005). *Prime Numbers: A Computational Perspective*. 2nd ed., Springer, New York.

Appendix 1. Catalogue of Goldbach Pairs Obtained

This appendix summarizes the principal Goldbach representations obtained during the computational program associated with the present study.

The examples are divided into three categories:

1. externally confirmed or certified prime-prime pairs;
2. strong probable-prime pairs;
3. difficult or incomplete searches in which no P/P event was found inside the explored region.

The status labels used below are deliberately conservative.

“Prime-prime confirmed” means that the user subsequently reported external confirmation of both primality conditions, either directly or through services such as FactorDB or dCode.

“PRP/PRP” means that both members passed the strong probable-prime stage used in the computational workflow, but no independent proof of primality is claimed here.

“Desert scan” means that no P/P event was found within the explored interval; it does not mean that no Goldbach representation exists.

A1.1. Confirmed Centered Prime-Prime Representations

A1.1.1. $E = 10^{1000} + 558$

Search mode: centered.

The successful centered deviation was $d = 2,069,958$. Thus $m = E/2$, $p = m - 2,069,958$, $q = m + 2,069,958$.

The user subsequently confirmed both p and q prime.

Status:

prime-prime confirmed.

This example belongs to the high-efficiency centered regime of the algorithm.

A1.1.2. $E = 10^{3000}$

Search mode: centered. The successful deviation was $d = 805,797$.

The pair is $p = 5 \times 10^{2999} - 805,797$, $q = 5 \times 10^{2999} + 805,797$.

The user confirmed both numbers prime.

Status:

prime-prime confirmed.

This is the largest fully confirmed centered pair in the principal dataset.

A1.2. Confirmed Peripheral Prime-Prime Representations

A1.2.1. $E = 10^{1366} + 772$

Search mode: peripheral.

$p = 40,939$, $q = 10^{1366} - 40,167$.

The user reported q prime through both FactorDB and dCode.

Since $p = 40,939$ is prime, $p + q = 10^{1366} + 772$.

Status: prime-prime confirmed.

A1.2.2. $E = 10^{1725} + 554$

Search mode: peripheral. $p = 73$, $q = 10^{1725} + 481$.

Indeed,

$73 + (10^{1725} + 481) = 10^{1725} + 554$.

The user externally confirmed q prime.

The small prime $p = 73$ was also selected early by an arithmetic priority criterion based on local gcd information.

Status:

prime-prime confirmed.

This is one of the most striking examples in the dataset because a very large even integer admits a representation with an exceptionally small first prime.

A1.2.3. $E = 10^{1750} + 444$

Search mode: peripheral.

$p = 27,191$, $q = 10^{1750} - 26,747$. The identity is $27,191 - 26,747 = 444$, hence $p + q = 10^{1750} + 444$.

The user confirmed q prime.

Status: prime-prime confirmed.

A1.2.4. $E = 10^{2500} + 8$

Search mode: peripheral.

$p = 25,391$, $q = 10^{2500} - 25,383$.

Since

$25,391 - 25,383 = 8$, the exact sum is $p + q = 10^{2500} + 8$.

The user reported q prime on dCode.

Status:

prime-prime confirmed.

This case is particularly important because the same E produced a long difficult centered search.

A1.2.5. $E = 10^{2750} + 6528$

Search mode: peripheral.

$p = 59,159$, $q = 10^{2750} - 52,631$.

Since

$59,159 - 52,631 = 6,528$, the pair satisfies $p + q = 10^{2750} + 6528$.

The user confirmed q prime on dCode.

Status: prime-prime confirmed.

A1.2.6. $E = 10^{2767} + 266$

Search mode: peripheral.

$p = 83,417$, $q = 10^{2767} - 83,151$.

Since

$83,417 - 83,151 = 266$, the exact identity is $p + q = 10^{2767} + 266$.

The user externally confirmed q prime.

Status: prime-prime confirmed.

A1.2.7. $E = 10^{3000} + 938$

Search mode: peripheral.

$p = 4,967$, $q = 10^{3000} - 4,029$.

Since

$4,967 - 4,029 = 938$, the pair satisfies $p + q = 10^{3000} + 938$.

The user reported q prime on FactorDB.

Status: prime-prime confirmed.

A1.2.8. $E = 10^{3500} + 96$

Search mode: peripheral.

A prime-prime peripheral representation was obtained and externally confirmed during the computational campaign.

Status:

prime-prime confirmed.

The complete explicit pair should be retained in the final archival

table if the original verification record is included with the supplementary material.

A1.2.9. $E = 10^{4000} + 44$

Search mode: peripheral.

A prime-prime peripheral representation was obtained and subsequently confirmed by the user.

Status:

prime-prime confirmed.

As for the preceding case, the exact archived pair should be inserted in the final supplementary dataset if the original verification record is appended.

A1.2.10. $E = 10^{6000}$

Search mode: peripheral.

$p = 11,213$, $q = 10^{6000} - 11,213$. Therefore, $p + q = 10^{6000}$.

The user independently confirmed q prime using both FactorDB and dCode.

Since $p = 11,213$ is prime, this is a confirmed prime-prime Goldbach representation.

Status:

prime-prime confirmed.

This is the largest confirmed example in the present dataset.

A1.3. Strong Probable-Prime Centered Representations

A1.3.1. $E = 10^{1000} + 2$

Search mode: centered. $m = 5 \times 10^{999} + 1$. The recorded deviation was $d = 2,304,238$. Thus $p = 5 \times 10^{999} - 2,304,237$, $q = 5 \times 10^{999} + 2,304,239$.

Both members were retained as strong probable primes in the computational workflow.

Status: PRP/PRP.

A1.3.2. $E = 10^{1500} + 2$

Search mode: centered. The recorded deviation was $d = 5,749,478$. The corresponding centered pair passed the strong probable-prime stage.

Status:

PRP/PRP.

A1.3.3. $E = 10^{2000} + 2$

Search mode: centered. The recorded deviation was $d = 6,313,132$. The pair is $p = 5 \times 10^{1999} - 6,313,131$, $q = 5 \times 10^{1999} + 6,313,133$.

Both members were retained as probable primes.

Status:

PRP/PRP.

A1.3.4. $E = 10^{2500} + 9248$

Search mode: centered.

The recorded successful deviation was $d = 1,507$. The midpoint is $m = 5 \times 10^{2499} + 4,624$. Thus $p = 5 \times 10^{2499} + 3,117$, $q = 5 \times 10^{2499} + 6,131$.

Both members were treated as strong probable primes in the experimental workflow.

Status:

PRP/PRP.

This is one of the easiest centered examples near 10^{2500} .

A1.3.5. $E = 10^{2600} + 722$

Search mode: centered. $m = 5 \times 10^{2599} + 361$. The successful deviation was $d = 1,582$.

Hence $p = 5 \times 10^{2599} - 1,221$, $q = 5 \times 10^{2599} + 1,943$.

Both members were retained as probable primes.

Status:

PRP/PRP.

This extremely small deviation contrasts sharply with the difficult centered cases near 10^{2500} .

A1.3.6. $E = 10^{700} + 324$

Search mode: centered.

The recorded successful deviation was $d = 1,424,901$.

The user confirmed the pair as PRP/PRP.

Status:

PRP/PRP.

A1.4. Additional large experimental pairs recorded earlier in the program

The wider computational program also produced several peripheral and non-central representations before the final adaptive protocol was fully standardized.

These examples are retained because they document the progression of the algorithm.

A1.4.1. $E = 10^{1100} + 974$

A representation was obtained with $p = 14,173$, $q = 10^{1100} - 13,199$.

Since

$14,173 - 13,199 = 974$, the identity $p + q = 10^{1100} + 974$ is exact.

This example was retained as a validated Goldbach candidate in the earlier computational series.

A1.4.2. $E = 10^{1100} + 976$

Several candidates were obtained. One recorded pair was based on $p = 782,083$, $q = 10^{1100} - 781,107$.

Additional recorded candidates included $p = 8,407,297$, $q = 10^{1100} - 8,406,321$, and $p = 12,838,993$, $q = 10^{1100} - 12,838,017$.

The large q values remained at probable-prime status in the recorded workflow.

These examples are therefore supplementary PRP-level candidates rather than certified pairs.

A1.5. Difficult centered searches and lower bounds

Negative or incomplete scans are important because they reveal the structure of difficult Goldbach regions.

They must not be interpreted as counterexamples.

A1.5.1. $E = 10^{2500} + 6$

Search mode: centered.

No verified double-PRP pair was found in the recorded scan extending beyond $d = 10,599,998$.

Thus, the computational experiment established only the scan statement: no P/P event detected within the explored centered interval.

It does not establish that $d_{\min}$ exceeds this value unless every admissible deviation below the bound has been exhaustively resolved under the same primality standard.

Status: difficult centered desert / incomplete global search.

A1.5.2. $E = 10^{2500} + 8$

Search mode: centered.

No verified double-PRP pair was found in the recorded centered scan extending beyond

$d = 6,099,999$.

However, the same E later yielded the peripheral confirmed pair $p = 25,391$, $q = 10^{2500} - 25,383$.

This example provides direct evidence that central difficulty $\neq$ global Goldbach difficulty.

A1.5.3. $E = 10^{4000} + 244$

This integer was used extensively in the Goldbach-desert phase of the project.

Many peripheral and constructed probable-prime pivots were examined.

A large number of $q = E - p$ candidates were found composite.

The experiment was especially useful for developing:

- p_{last} tracking;
- survivor resistance;
- local PRP-gap measurements;
- sieve-first methodology;
- distinction between primality deserts and Goldbach deserts.

The case should be regarded primarily as a methodological development dataset rather than as one of the final successful examples.

A1.6. Comparison Table

E	Mode	Parameter	Pair status	Main significance
$10^{700} + 324$	Centered	$d = 1,424,901$	PRP/PRP	Early centered large-scale test
$10^{1000} + 2$	Centered	$d = 2,304,238$	PRP/PRP	Large centered pair
$10^{1000} + 558$	Centered	$d = 2,069,958$	Prime-prime confirmed	Certified centered case
$10^{1100} + 974$	Peripheral	$p = 14,173$	Recorded validated pair	Earlier peripheral series
$10^{1500} + 2$	Centered	$d = 5,749,478$	PRP/PRP	Centered scalability
$10^{1725} + 554$	Peripheral	$p = 73$	Prime-prime confirmed	Extremely small peripheral p
$10^{1750} + 444$	Peripheral	$p = 27,191$	Prime-prime confirmed	MAX-SIEVE FIRST success
$10^{2000} + 2$	Centered	$d = 6,313,132$	PRP/PRP	Upper high-efficiency centered range
$10^{2500} + 8$	$> 6,099,999$	Centered scanned	No P/P in scanned region	Central desert
$10^{2500} + 8$	Peripheral	$p = 25,391$	Prime-prime confirmed	Central/peripheral contrast
$10^{2500} +$	Centered	$d = 1,507$	PRP/PRP	Very easy centered case

A1.7. Verification Policy

The computational record uses several levels of verification.

Level 1: Modular Survivor

A candidate survives all applied small-prime sieves.

No primality claim is made.

Level 2: Probable Prime

The candidate passes a strong probable-prime procedure such as repeated Miller–Rabin or a Baillie–PSW-type test.

The candidate is reported as PRP.

Level 3: Externally Confirmed Prime

The user reports independent verification through a dedicated external database or primality service.

These cases are reported here as confirmed according to the external verification record.

Level 4: Primality Certificate

When a formal ECPP or equivalent certificate is available, the number should be labeled certified prime.

The final archived version of the dataset should preserve the exact

method used for each member of each pair.

A1.8. Arithmetic Consistency Checks

Every final pair must satisfy four elementary checks.

First, $p + q = E$ must hold exactly.

Second, the decimal exponent must match the intended E .

Third, both p and q must be odd except for the trivial representations involving 2.

Fourth, the claimed status must match the strongest verification actually performed.

These checks are especially important when compact expressions such as

$10^n - a$ or

$5 \times 10^{(n-1)} \pm a$ are used.

A1.9. Principal Empirical Pattern

The catalogue reveals three recurring types of successful behavior. The first is a centered moderate-deviation pair, as observed for $E = 10^{1000} + 2$ and $E = 10^{3000}$.

The second is an exceptionally small centered deviation, as observed for $E = 10^{2500} + 9248$ and $E = 10^{2600} + 722$.

The third is a small peripheral prime, as observed for $E = 10^{1725}$

+ 554, $E = 10^3000 + 938$, and $E = 10^6000$.

The coexistence of these regimes is one of the main reasons the final algorithm uses an adaptive multi-zone strategy rather than a single fixed search geometry.

A1.10. Final Note

This appendix is intended as an experimental record.

The examples demonstrate that the adaptive algorithm can construct Goldbach candidates aAppendix 2. Best Current Reduction Toward a Proof

This appendix summarizes the strongest theoretical reduction obtained in the parallel proof-oriented part of the project.

It is included because the computational experiments repeatedly indicate that the principal difficulty appears after modular admissibility has already been resolved. The theoretical reduction arrives at a closely related conclusion: the remaining obstruction is not the production of admissible candidates, but a uniform cancellation problem controlling the parity-sensitive part of the Goldbach count.

This appendix does not claim a proof of Goldbach's conjecture.

The objective is to identify as precisely as possible the final analytic statement that would be sufficient to complete the argument.

A2.1. Starting Point

Let E be a sufficiently large even integer.

The binary Goldbach problem asks whether there exist primes p and q such that $p + q = E$.

Equivalently, one seeks a prime p for which $q = E - p$ is also prime. The proof program considered here begins from a weighted counting function over primes p .

Write schematically

$R(E)$ = weighted number of prime-prime representations of E .

The objective is to establish $R(E) > 0$.

A positive value immediately gives at least one Goldbach representation.

A2.2. Chen Support

Chen's theorem provides a powerful approximation to the binary problem.

For every sufficiently large even integer E , there exist many representations $E = p + n$

in which p is prime and n belongs to $P \cap P_2$,

where P denotes primes and P_2 denotes integers having exactly two prime factors, counted with multiplicity under the usual convention.

This produces a positive weighted Chen count.

Denote this quantity by $C(E)$.

Schematically,

$C(E)$ = contribution from $q \in P \cap P_2$.

The important point is that $C(E)$ has the correct Goldbach scale: $C(E) \gg E / (\log E)^2$ for sufficiently large even E , after suitable weighting and normalization.

This part provides positive mass. However, it does not distinguish between q prime and q semiprime.

That distinction is exactly the parity problem that remains.

A2.3. Splitting Prime and Semiprime Contributions

On the Chen support $q \in P \cap P_2$, introduce a parity-sensitive weight capable of distinguishing $E \in U$ primes from products of two primes. Let

$R(E)$ denote the contribution of genuine prime-prime representations, and let $S_2(E)$ denote the contribution from prime-semiprime representations.

Then

$C(E) = R(E) + S_2(E)$.

The objective is to separate these two terms.

A Liouville-type sign is natural because, on squarefree integers, $\lambda(q) = -1$ for q prime, whereas $\lambda(q) = +1$ for q equal to a product of two primes.

After choosing the sign convention consistently, the weighted parity difference can be written as a quantity $L(E)$.

The resulting algebraic relation is $R(E) = (C(E) - L(E))/2$.

This is the key identity in the reduction. Therefore, a sufficient condition for Goldbach is $C(E) > L(E)$.

More robustly, it is enough to prove $|L(E)| = o(C(E))$.

Since

$C(E) \gg E/(\log E)^2$, it would suffice to establish $L(E) = o(E/(\log E)^2)$ uniformly for all sufficiently large even E .

A2.4. Liouville–Möbius expansion

The Liouville function can be expanded using the identity $\lambda(n) = \sum_{d^2 \mid n} \mu(n/d^2)$, where μ is the Möbius function.

Substituting this into the parity term gives a decomposition of $L(E)$ into square-divisor layers.

Schematically,

$L(E) = \sum_p w_E(p) \lambda(E - p)$, and therefore

$L(E)$

$\sum_p w_E(p) \sum_{d^2 \mid E-p} \mu((E-p)/d^2)$.

Interchanging the order of summation gives

$L(E)$

$\sum_d$

$\sum_p \text{with } d^2 \mid E-p$

$w_E(p) \mu((E-p)/d^2)$.

The $d = 1$ term is

$S(E) = \sum_p w_E(p) \mu(E - p)$.

The terms with $d \geq 2$ correspond to square divisibility.

Under the natural size restrictions of the Chen support and standard square-divisor estimates, these higher-square contributions are expected to be lower order relative to $E/(\log E)^2$.

Thus the principal parity obstruction reduces to the Möbius correlation $S(E) = \sum_p w_E(p) \mu(E - p)$.

A2.5. The Decisive Uniform Estimate

The strongest reduction obtained in the project is the following sufficient condition.

For every $\epsilon > 0$, uniformly for all sufficiently large even E ,

$|\sum_p w_E(p) \mu(E - p)|$

$\leq$

$\epsilon E/(\log E)^2$.

This condition is referred to in the proof program as C6.

Its essential feature is uniformity in E .

An estimate holding on average over E is not sufficient for the binary Goldbach conjecture.

An estimate holding for almost all even E is also not sufficient.

The conjecture requires positivity for every sufficiently large even E .

Therefore, the critical target is the pointwise uniform statement:

C6-uniform:

for every $\epsilon > 0$, there exists $E_0(\epsilon)$ such that for every even $E \geq E_0(\epsilon)$, $|S(E)| \leq \epsilon E / (\log E)^2$.

A2.6. Why C6 Would Imply Goldbach

Assume C6-uniform.

Then the principal Möbius term satisfies $S(E) = o(E / (\log E)^2)$ uniformly in E .

The square-divisor layers with $d \geq 2$ are lower order under the reduction.

Therefore,

$$L(E) = o(E / (\log E)^2).$$

On the other hand, the Chen contribution satisfies $C(E) \ll E / (\log E)^2$.

Hence, for sufficiently large E , $|L(E)| < C(E)$.

Using

$$R(E) = (C(E) - L(E))/2, \text{ we obtain } R(E) > 0.$$

Therefore, every sufficiently large even E has a prime-prime representation.

The remaining finite set of smaller even integers could then be covered by existing computational verification.

Thus C6-uniform would imply the binary Goldbach conjecture.

A2.7. Why This is Not Yet a Proof

The logical chain above is conditional.

The missing step is precisely the uniform Möbius cancellation

$$|S(E)| = o(E / (\log E)^2) \text{ for every sufficiently large even } E.$$

This estimate has not been proved in the present work.

Consequently, the statement $R(E) > 0$ has not been established unconditionally by this method.

The reduction should therefore be viewed as a localization of the remaining difficulty rather than a completed proof.

A2.8. Relation to the Parity Barrier

The structure of C6 is consistent with the classical parity obstruction in sieve theory.

The sieve can efficiently separate numbers with small prime factors from numbers without small prime factors.

It can also produce the Chen support

$P \cap P_2 \cap U$

However, distinguishing systematically between one prime factor and two prime factors requires genuinely parity-sensitive information.

The Möbius or Liouville function carries precisely this information.

Thus, the appearance of the Möbius correlation is not accidental.

It is a reformulation of the final parity-sensitive step.

This also explains why repeated improvements of purely local sieve density did not close the proof.

A2.9. Connection with the Computational Algorithm

The computational and theoretical sides of the project developed independently but converged toward a similar structural picture.

The algorithm first removes local modular obstructions.

After MAX-SIEVE FIRST, many surviving candidates remain highly admissible.

Yet difficult cases can still produce long sequences without a P/P event.

This was described experimentally as a synchronization desert.

The theoretical reduction similarly shows that the remaining obstruction after Chen-type admissibility is a signed correlation involving $\mu(E - p)$.

Thus, both viewpoints separate the problem into two stages:

Stage 1:

local admissibility and sieve structure; Stage 2:

parity-sensitive prime synchronization.

The experiments do not prove C6.

However, they provide a computational interpretation of why a parity-sensitive term naturally remains after strong sieving.

A2.10. Almost-all Versus all-E Statements

A major point in the proof program was the distinction between three levels of possible estimates.

Level 1: Averaged Estimate

One may be able to show cancellation after averaging over many E .

This can provide information on the global distribution of exceptional cases.

It does not imply Goldbach for each individual E .

Level 2: Almost-All Estimate

One may establish that $R(E) > 0$ for all but a sparse exceptional set of even integers.

This is much stronger computationally and analytically.

However, an arbitrarily thin infinite exceptional set is still incompatible with a proof of Goldbach.

Level 3: Uniform Pointwise Estimate

The required terminal statement is $R(E) > 0$ for every sufficiently large even E .

This is why C6 must be uniform.

The transition from almost-all control to uniform control is the principal unresolved step.

A2.11. Earlier Routes Explored

Several analytic transformations were considered during the proof program.

These included:

- character decompositions;
- Gauss sums;
- Kloosterman-type transforms;
- averaging over varying moduli;
- Fourier decompositions of admissible residue sets;
- bilinear and multilinear decompositions;
- Type I, Type II, and Type III structures;

- spectral formulations;
- reciprocal-fraction kernels;
- inter-modulus cancellation.

These approaches were useful for clarifying the structure of the obstruction.

However, none produced the required uniform pointwise gain without either losing the crucial cancellation or reconstructing an expression essentially equivalent to the original parity problem.

Several routes failed because absolute values or Cauchy–Schwarz were applied too early, destroying oscillation across moduli.

Other routes reduced the problem to estimates of comparable difficulty to C6.

These failures are informative because they strongly suggest that any successful continuation must preserve signed cancellation until very late in the argument.

A2.12. Gauss-Character Stage

One advanced version of the reduction led to expressions schematically of the form $\sum_{\text{sum over characters } \chi \bmod c} \tau_A(\chi)/\phi(c)$

$A_{\{\chi, c\}} B_{\{\chi, c\}}$. Here, $\tau_A(\chi)$ represents a Gauss-type factor, $A_{\{\chi, c\}}$ and $B_{\{\chi, c\}}$ represent arithmetic transforms, and γ_c is a modulus-dependent coefficient.

The essential strategic requirement was to retain oscillation across c .

Taking absolute values independently for each modulus destroyed too much cancellation.

Similarly, applying Cauchy–Schwarz before opening the Gauss factor tended to lose the intermodulus structure.

The remaining question was whether opening the Gauss sum and the c -average simultaneously could reveal genuine cancellation across moduli. No complete uniform estimate emerged from this route.

A2.13. The terminal Möbius form

After several transformations, the proof program repeatedly returned to the same basic obstacle: a Möbius-weighted prime correlation at the Goldbach scale.

This recurrence is one reason the final reduction is formulated directly as C6.

Rather than introduce additional notation that eventually returns to the same correlation, the current position is to treat

$S(E) = \sum_{\text{sum over } p} w_E(p) \mu(E - p)$
as the true terminal object.

Any future proof attempt should therefore be judged by one criterion: does it produce a genuinely uniform pointwise gain for $S(E)$?

If not, it has not crossed the final barrier.

A2.14. Minimal Sufficient Theorem

A useful way to state the remaining target is as a minimal sufficient theorem.

Uniform Möbius–Goldbach Cancellation Theorem

For every $\epsilon > 0$, there exists E_0 such that, for every even $E \geq E_0$,

$$\left| \sum_{\text{sum over } p} w_E(p) \mu(E - p) \right| \leq \epsilon E / (\log E)^2.$$

If this theorem is proved with the weights required by the Chen decomposition, then the binary Goldbach conjecture follows.

The strength of this formulation is that it is narrower than Goldbach itself.

It does not directly assert the existence of a prime q .

It asserts cancellation of a signed arithmetic correlation.

However, it remains a deep pointwise statement and is not currently established.

A2.15. Necessary Audit of any Future Proof Route

Any proposed continuation should pass several logical checks.

First, the new lemma must not already contain the conclusion $R(E) > 0$ in disguised form.

Second, it must be strictly weaker than Goldbach.

Third, it must not simply assume a positive density of P/P pairs.

Fourth, it must survive extreme Chen-support configurations, including cases dominated by P2 contributions.

Fifth, it must not use an averaging argument that leaves an uncontrolled exceptional set. Sixth, it must preserve enough cancellation to reach the scale $o(E/(\log E)^2)$.

If a proposed lemma fails any of these audits, it should not be treated as progress toward the proof.

A2.16. Possible Connection with Conditioned Admissible Sets

The computational experiments suggest one possible reformulation for future work.

Instead of studying the Möbius correlation over the entire prime support directly, one may condition first on residue classes that survive a deep modular sieve.

Schematically,

$S(E)$

$\sum_{\text{sum over admissible classes } a} \sum_{\text{sum over } p \text{ in class } a} w_E(p) \mu(E - p).$

The motivation is that the small-prime information has already been explicitly extracted.

The remaining sum then measures cancellation only among the resistant survivors.

If a uniform cancellation estimate could be established after this conditioning, without reintroducing the parity problem in equivalent form, it could provide a new route toward C6.

This possibility remains speculative.

No theorem of the required strength is claimed here.

A2.17. Why the Logarithmic Central-Window Conjecture Is Not Used

The computational study also investigated the heuristic windows $H_2(E) = (\log E)^2$, $H_3(E) = (\log E)^3$, and

$$H\infty(E) = (\log E)^3 \log \log E.$$

A theorem asserting the existence of a prime-prime pair inside H_3 for every sufficiently large E would directly imply Goldbach.

However, no such theorem has been established.

Therefore, the logarithmic-window observations are not used in the proof reduction.

The present appendix instead follows the Chen–Liouville–Möbius route because it isolates an analytic cancellation statement without assuming a central P/P event.

A2.18. What has Been Achieved

The theoretical work has achieved the following.

1. The Goldbach positivity problem is expressed through a weighted count $R(E)$.
2. Chen's theorem supplies a positive quantity $C(E)$ at the correct scale.
3. The prime/semiprime distinction is isolated through a parity-sensitive term $L(E)$.
4. The exact relation $R(E) = (C(E) - L(E))/2$ reduces the problem to controlling $L(E)$.
5. The Liouville identity converts $L(E)$ into square-divisor Möbius sums.
6. The principal obstruction is reduced to $S(E) = \sum \text{over } p \text{ of } w_E(p) \mu(E - p)$.
7. The remaining sufficient estimate is the uniform C_6 bound $|S(E)| = o(E/(\log E)^2)$.

This is a substantial reduction of the logical structure.

It is not the completion of the proof.

A2.19. What Remains Open

The unresolved point is singular and precise. One must obtain enough cancellation in $\sum \text{over } p \text{ of } w_E(p) \mu(E - p)$ uniformly for every sufficiently large even E .

The estimate must be pointwise in E .

It must retain the correct Goldbach-scale normalization.

It must not lose the necessary power through absolute values, early Cauchy–Schwarz, or averaging over E .

It must also remain valid under the exact support and weights inherited from the Chen decomposition.

No argument developed so far satisfies all these requirements simultaneously.

A2.20. Conditional Completion of the Proof Chain

The logical implication can finally be summarized as

Chen positive mass

→

$$C(E) \gg E/(\log E)^2$$

→

parity decomposition

→

$$R(E) = (C(E) - L(E))/2$$

→

Liouville–Möbius expansion

→

$$L(E) = S(E) + \text{lower-order square terms}$$

→

C_6 -uniform

→

$$L(E) = o(E/(\log E)^2)$$

69

→

$$R(E) > 0$$

→

Goldbach for all sufficiently large even E

→

finite verification for the remaining E .

The only unproved arrow of decisive size is the C_6 -uniform cancellation step.

A2.21. Final Statement

The present proof program has therefore not demonstrated Goldbach's conjecture.

Its strongest outcome is a conditional terminal reduction.

The remaining obstruction is no longer described vaguely as “the parity problem” but as a concrete uniform Möbius–prime correlation estimates at the Goldbach scale.

This formulation provides a precise benchmark for future work.

Any genuinely new proof mechanism must ultimately produce a uniform gain strong enough to establish C_6 , or replace C_6 with an alternative statement of strictly weaker or comparable strength that still forces $R(E) > 0$.

Until such a step is proved, the result remains a reduction and not a proof. confirmed representations at very large decimal scales.

They do not establish a universal upper bound for the first Goldbach pair, nor do they imply that every even integer of comparable size can be solved with the same computational effort.

The distinction between empirical success, probable-prime evidence, external confirmation, and theorem-level proof is maintained throughout.

Appendix 3. How to Run the Algorithm in Its Optimal Current Form

This appendix gives the recommended operational version of the adaptive Goldbach algorithm used in the present study.

The objective is not to test candidates blindly, but to reduce the search space as aggressively as possible before expensive primality testing.

A3.1. Input

Choose an even integer E . Compute $m = E/2$.

The algorithm then works with either $p = m - d$, $q = m + d$ in centered mode, or $q = E - p$ in peripheral mode.

A3.2. Initial Audit

Before testing any large candidate, perform a structural audit of E and m .

Record:

- parity;
- decimal length;

- small prime factors of E when easily available;
- residues of E modulo small primes;
- residues of m modulo small primes;
- local admissibility restrictions;
- expected survivor density;
- any obvious residue-class asymmetry.

This initial audit determines which search region is likely to be cheapest.

A3.3. Choose the Initial Zone

Three zones are available:

1. peripheral region;
2. centered region;
3. intermediate region near a square-root-type scale.

For a pure first-hit search at extreme size, peripheral mode is often the best first choice because p is already a known prime and only q needs a large primality test.

For studies of central Goldbach behavior, start at the center.

Do not assume that the same region is optimal for every E.

A3.4. Peripheral Mode

Generate small or moderate primes p. For each p, define $q = E - p$. Before any strong primality test, reject p whenever q is divisible by one of the sieve primes.

Continue until only a small premium survivor population remains. The peripheral mode is particularly efficient because p does not need to be tested again.

A3.5. Centered Mode

Generate deviations d.

For each small sieve prime r, remove values of d satisfying $m - d \equiv 0 \pmod r$ or $m + d \equiv 0 \pmod r$.

The remaining deviations form the admissible centered set.

Search can initially be organized around

$H_2 = (\log E)^2$, $H_3 = (\log E)^3$, and

$H_\infty = (\log E)^3 \log \log E$.

These are search scales, not universal proved bounds.

A3.6. Apply MAX-SIEVE FIRST

Do not move immediately from a shallow sieve to Miller–Rabin.

Increase the sieve depth as far as practical.

A typical sequence may be

100,

1000,

10,000,

100,000, 1,000,000, or deeper if computationally affordable.

The exact thresholds depend on E and the implementation.

The principle is invariant:

cheap rejection first, expensive testing last.

A3.7. Survivor Ranking

Track which candidates survive each sieve depth.

A candidate surviving deeper levels has higher resistance.

Rank candidates by:

- maximum sieve depth survived;
- local survivor isolation;

- favorable residue structure;
- low predicted testing cost;
- optional arithmetic priority scores.

Only the strongest survivors should reach the final primality-testing stage.

A3.8. Test the Cheapest Side First

When both p and q are large, test first the side that is likely to fail most cheaply.

If the first side is composite, stop immediately for that candidate.

Only test the second side when required.

This rule is essential in centered searches, where both numbers may contain thousands of digits.

A3.9. Strong Probable-Prime Testing

For premium survivors, use a strong probable-prime protocol.

A practical sequence is:

1. deep modular filtering;
2. several Miller–Rabin rounds;
3. Baillie–PSW-type testing when available;
4. final independent certification for selected results.

Never label a PRP result as certified prime.

A3.10. First-Hit Mode

For very large E, first-hit mode should normally be preferred.

Stop as soon as a reliable P/P pair is found.

Do not continue merely to identify a smaller d unless the experiment specifically requires $d_{\min}$.

This greatly reduces total runtime.

A3.11. Exact $d_{\min}$ Mode

If the objective is the smallest centered deviation, every admissible deviation below the reported d must be resolved.

This mode is much more expensive.

It should therefore be used only when the mathematical question specifically concerns central minimality.

A3.12. Record the State Chain

During difficult searches, record local states:

C/C,

C/P,

P/C, P/P.

Track:

- lengths of C/C runs;
- transitions between states;
- first one-sided prime events;
- rank of the first P/P event;
- gaps between premium survivors.

These quantities are useful for identifying synchronization deserts.

A3.13. Detect Cold Regions Early

Do not continue indefinitely through a long cold chain.

If a region produces many premium survivors but no P/P event, classify it as locally cold.

Then choose among:

- deepen the sieve;

- jump forward;
- change zone;
- activate CHIRURGIE;
- parallelize survivor testing.

The purpose of adaptation is to prevent the strongest primality tests from being wasted repeatedly in a region that is empirically unproductive.

A3.14. CHIRURGIE Mode

Activate CHIRURGIE immediately when the search becomes unusually difficult.

Measure:

- raw candidate count;
- survivor count at each sieve depth;
- survivor density;
- mean survivor spacing;
- maximum survivor gap;
- number of premium survivors;
- C/C chain length;
- frequency of P/C and C/P;
- asymmetry between the two sides;
- distances to neighboring primes when feasible.

Then decide whether the difficulty is mainly: admissibility-limited, synchronization-limited, or mixed.

This diagnosis determines the next algorithmic move.

A3.15. Adaptive Zone Switching

If the center is cold, try the periphery.

If the periphery produces a long chain of failures, return to the center or test an intermediate region.

If one region suddenly produces P/C or C/P states more frequently, exploit that region more heavily.

The algorithm should behave as a dynamic search system, not as a fixed scan.

A3.16. Maintain p_{last} and Search Bounds

For every peripheral search, record the last tested or last fully confirmed prime p.

For every centered search, record the last fully resolved deviation d.

This prevents accidental repetition and allows desert bounds to be reported correctly.

A bound is valid only if the corresponding interval has actually been exhaustively resolved.

A3.17. Final Certification

Once a P/P candidate is found, perform the final audit.

Check:

E exactly; p exactly; q exactly; $p + q = E$ exactly; decimal lengths; parity; probable-prime status; certification status.

For important records, obtain independent primality verification or a formal certificate where possible.

A3.18. Recommended Operating Strategy by Size

For approximately $E \leq 10^{2000}$, center and periphery can both be

explored relatively freely.

For approximately

$10^{2000} < E \leq 10^{3000}$, prefer deeper sieving and first-hit mode.

For

$E > 10^{3000}$, use the periphery early unless the experiment specifically concerns central structure.

At this scale, minimizing the number of final strong tests becomes more important than increasing the number of raw candidates.

These are empirical recommendations, not theoretical limits.

A3.19. Compact Operational Algorithm

The optimal practical protocol is:

Audit E.

Choose zone.

Generate candidates.

MAX-SIEVE FIRST.

Rank survivors.

Test the cheapest side first.

Apply strong tests only to premium survivors.

Record chain structure.

Detect cold regions.

Activate CHIRURGIE when needed.

Jump or switch zone adaptively.

Stop at first reliable P/P.

Audit $p + q = E$.

Record status and search bound.

This is the current best-performing version of the algorithm developed in the project.

A3.20. Final Practical Principle

The algorithm should not be understood as a method for testing more numbers.

Its purpose is to test fewer numbers, but better ones.

The most important optimization is therefore not raw speed.

It is selective pressure.

The stronger the sieve, the better the survivor ranking, and the faster the reaction to cold regions, the fewer several-thousand-digit primality tests are required.

That principle is the core of the present algorithm.

Global Closing Statement

This study began with a simple computational question: how can one find a Goldbach pair for an isolated even integer so large that direct search becomes impractical?

The answer developed progressively.

The first important step was to move away from blind testing and to organize the search around structure. Centered deviations made it possible to study the neighborhood of $E/2$ through $p = E/2 - d$, $q = E/2 + d$.

Peripheral search then showed that the center is only one part of the problem and that very large even integers can sometimes possess remarkably small successful primes.

This led to an adaptive strategy rather than a single fixed search rule.

The second major step was MAX-SIEVE FIRST.

The experiments showed repeatedly that the sieve remains effective even when E contains several thousand digits. The practical difficulty does not come primarily from generating candidates. It comes from deciding which survivors deserve a costly primality test.

This observation changed the algorithm fundamentally.

Instead of testing candidates as soon as they appear, the method now applies strong modular selection first, ranks the survivors, and reserves the most expensive tests for a small premium population. The third discovery came from the difficult cases.

Long central deserts initially seemed likely to be caused simply by a low density of admissible deviations. Detailed comparisons showed that this explanation was insufficient. Some difficult cases retained many admissible survivors but still failed to produce simultaneous primality over extremely long ranges.

This led to the distinction between admissibility geometry and prime synchronization.

That distinction is one of the central conceptual outcomes of the work.

A Goldbach desert can be produced not only because candidates are rare, but because many good candidates fail to become prime on both sides at the same time.

The resulting synchronization-desert picture explains why a relatively small 2500-digit case can be extremely difficult while a favorable 6000-digit case can be solved rapidly.

The experimental campaign also produced a broad collection of explicit pairs, including centered and peripheral examples across several thousand decimal digits and a confirmed representation at $E = 10^6000$.

These results do not prove Goldbach's conjecture, but they show that constructive Goldbach search remains feasible at scales far beyond ordinary direct enumeration.

The work therefore brings three things to Goldbach research.

First, it provides a practical large-integer search architecture.

Second, it provides an experimental language for distinguishing local modular difficulty from latestage prime synchronization.

Third, it connects this computational picture with the parallel theoretical reduction.

The proof-oriented work reaches a strikingly similar boundary.

Chen-type arguments provide positive mass and local admissibility, but the remaining step is parity-sensitive. After the Liouville–Möbius reduction, the unresolved obstruction becomes a uniform cancellation estimate of the form

$$|\sum_{p \leq E} w_p \mu(E - p)|$$

$$o(E/(\log E)^2) \text{ for every sufficiently large even } E.$$

This is the missing step.

The present experiments do not prove this estimate.

They do, however, help explain why it appears: local sieving can remove an enormous number of obstructions, yet it does not force simultaneous primality.

Thus the computational and theoretical investigations converge on the same conclusion.

The unresolved difficulty lies after admissibility.

What remains for a proof is not another deeper local sieve, nor another larger numerical experiment, but a genuinely uniform mechanism controlling the parity-sensitive correlation that survives all those local reductions.

That is the precise point at which the present work stops.

The result is therefore neither a proof nor merely a collection of examples.

It is a connected program: from centered logarithmic windows, to adaptive peripheral search, to MAX-SIEVE FIRST, to Goldbach deserts, to prime synchronization, to the final uniform Möbius barrier.

The numerical experiments reveal where the algorithm succeeds.

The difficult cases reveal why it sometimes slows.

And the theoretical reduction identifies what would still have to be proved to cross the final boundary from computational evidence to a theorem valid for every even integer.

That connection is, in our view, the principal contribution of this work.