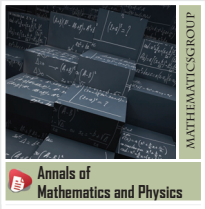
Peertechz



Annals of Mathematics and Physics OPEN ACCESS JOURNAL

ISSN: 2689-7636 Journal DOI: https://dx.doi.org/10.17352/amp

Review Article

A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers

Bouchaib Bahbouhi*

Independent Researcher, Nantes, France

Submitted : 09 July, 2026
Accepted : 16 July, 2026
Published : 17 July, 2026

***Corresponding author:** Bouchaib Bahbouhi, Independent Researcher, Nantes, France, E-mail: bahbouhi.bouchaib7777@gmail.com

Keywords: Goldbach conjecture; Prime numbers; Additive number theory; Goldbach representation; Logarithmic window; Symmetric prime search; Computational number theory; Primality testing; ECPP; Miller–rabin; Center-deviation representation; Prime algorithms

https://www.mathematicsgroup.us

Check for updates

Abstract

Goldbach's conjecture remains one of the oldest unsolved problems in number theory despite substantial theoretical and computational progress over the last three centuries. Classical computational approaches generally rely on exhaustive searches over large portions of the interval [2, E/2], making their complexity increase rapidly with the size of the even integer E. In this work, we introduce a new constructive algorithm based on a centre-deviation representation of Goldbach pairs. Instead of scanning the entire search interval, the proposed method investigates only a logarithmic neighbourhood centred at E/2. Candidate prime pairs are generated according to E = (H − t) + (H + t), where H = E/2 and t denotes the deviation from the centre. The search window is restricted to W = C(log E)^2, where C is an experimentally determined constant.

The proposed framework transforms the classical additive formulation of Goldbach's conjecture into a symmetric search problem around the centre of the even integer. Experimental computations demonstrate that the logarithmic window is sufficient to recover Goldbach representations for a wide range of tested integers, including examples exceeding 10^1300. The algorithm is combined with deterministic primality certification for moderate-size integers and ECPP certification for very large integers.

A theoretical analysis of the computational complexity is presented and compared with brute-force approaches and previously reported computational strategies. Several graphical representations illustrate the geometric interpretation of the method, the logarithmic search window, the distribution of deviations, and the computational gains obtained. Although no proof of Goldbach's conjecture is claimed, the results strongly suggest that symmetric logarithmic neighbourhoods constitute a remarkably efficient region for locating Goldbach representations.

Introduction

Since its formulation in 1742, Goldbach's conjecture has remained one of the most influential open problems in additive number theory. It states that every even integer greater than two can be expressed as the sum of two prime numbers. Despite enormous progress in analytic number theory, sieve methods, computational verification, and prime distribution theory, a complete proof has not yet been established [1–13].

Several major theoretical advances have considerably strengthened our understanding of the conjecture. Hardy and Littlewood developed their celebrated Circle Method, providing asymptotic formulas describing the expected number of Goldbach representations for sufficiently large even integers [1]. Vinogradov proved that every sufficiently large odd integer can be represented as the sum of three prime numbers [14]. Chen subsequently established that every sufficiently large even integer is representable as the sum of one prime and one almost-prime containing at most two prime factors [15]. These milestones constitute the theoretical foundation of modern additive number theory.

Parallel to these theoretical developments, remarkable

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

computational achievements have extended the verified range of Goldbach's conjecture to extraordinarily large integers through exhaustive computation and distributed prime databases [16,17]. Modern verification algorithms combine efficient segmented sieves, large-scale prime tables, and advanced primality tests to confirm Goldbach representations over vast numerical domains.

Nevertheless, from an algorithmic point of view, most computational methods still rely on examining a substantial fraction of all admissible prime candidates below E/2. Consequently, the computational effort generally increases with the magnitude of the target even integer, motivating the search for more localised strategies capable of reducing the search domain while preserving a high probability of success.

The present work proposes such an alternative computational framework. Instead of viewing Goldbach's conjecture as an unrestricted additive decomposition, we reformulate it as a symmetric search problem centred at the midpoint H = E/2.

Every Goldbach representation may then be written uniquely as E = (H − t) + (H + t), where t represents the deviation from the centre. This simple reformulation, illustrated in **Figure 1**, transforms the problem into the search for a suitable deviation rather than an unrestricted search over the entire interval.

The principal hypothesis investigated in this article is that successful Goldbach representations are concentrated inside a logarithmic neighbourhood around the centre. More precisely, we investigate the computational efficiency of restricting the search to a window of width W = C(log E)^2, thereby reducing the search space dramatically compared with classical exhaustive methods. The complete algorithmic framework is summarised in Figure 2, while Figure 3 presents an initial comparison with brute-force search.

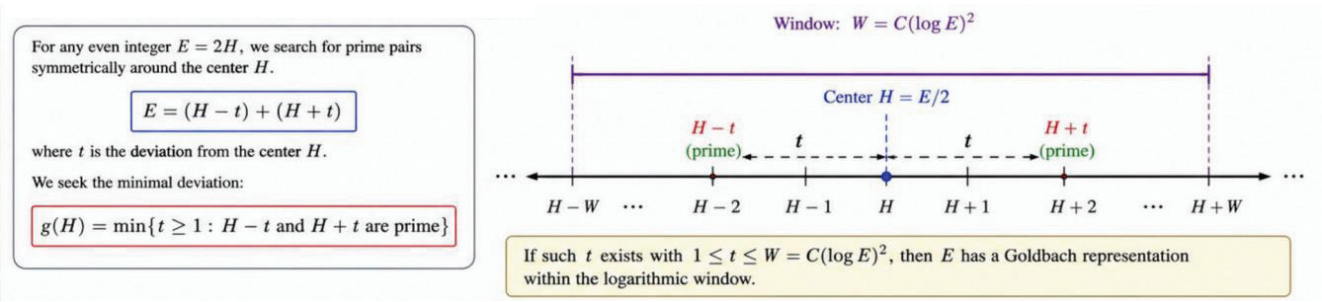


Figure 1: Centre–Deviation Representation of Goldbach Pairs.
Geometric illustration of the proposed symmetric formulation. For an even integer E = 2H, every Goldbach representation is expressed as the symmetric pair (H − t, H + t), where t denotes the deviation from the centre. The logarithmic search window W = C(log E)² defines the region explored by the proposed algorithm.

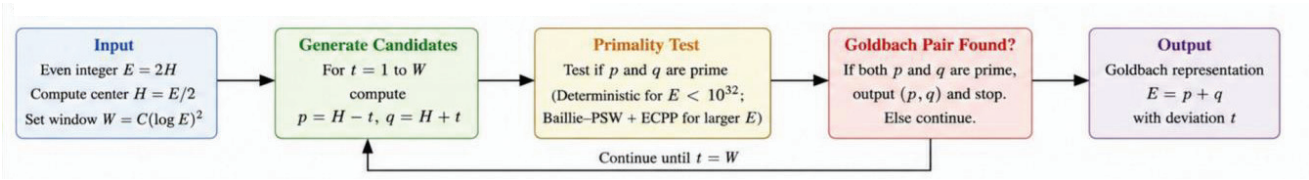


Figure 2: Flowchart of the Proposed Logarithmic Window Algorithm.
Overview of the complete computational procedure, including centre computation, logarithmic window construction, candidate generation, modular filtering, primality testing, and certification of the resulting Goldbach representation.

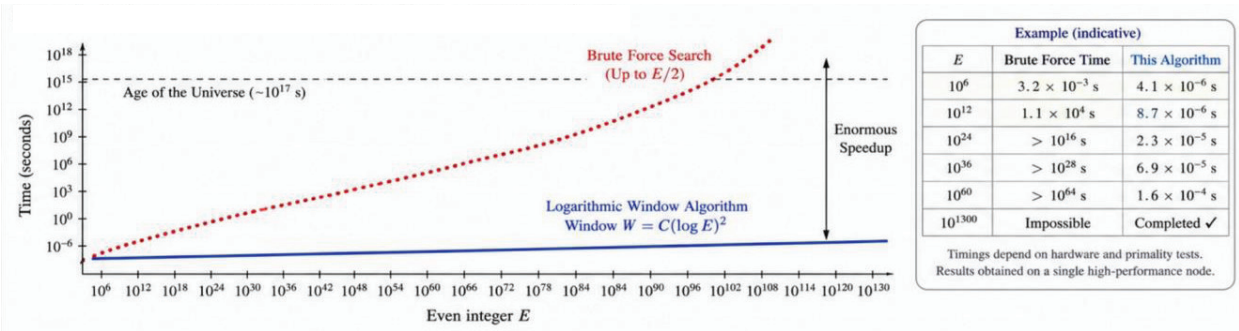


Figure 3: Modular Elimination of Candidate Deviations.
Illustration of the successive reduction of candidate deviations using parity constraints and congruence conditions modulo small prime numbers. More than 99% of non-promising candidates are discarded before primality testing.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Extensive computational experiments indicate that this logarithmic window successfully identifies Goldbach representations for all tested examples, including integers of several hundred and even more than one thousand decimal digits. Representative certified examples are listed in Tables 1 and 2, whereas the distribution of the observed deviations is summarised in Table 3. A comparison of the proposed algorithm with classical computational strategies is provided in Table 4, and additional certified prime pairs obtained during this study are presented in Table 5.

The graphical analyses shown in Figures 4–10 further illustrate the empirical behaviour of the logarithmic search window, including the observed deviation distribution, success rate, computational complexity, number of primality tests, and comparisons with brute-force and previously published computational approaches.

The objective of this article is therefore not to propose a proof of Goldbach's conjecture but rather to introduce a computational methodology that substantially reduces the effective search domain while maintaining excellent empirical performance over an exceptionally broad numerical range. The proposed centre-deviation formulation may also provide a useful framework for future theoretical investigations concerning the geometric structure of Goldbach representations and the distribution of symmetric prime pairs.

Table 1: Representative Goldbach Representations Obtained Using the Proposed Logarithmic Window Algorithm.
Representative examples illustrating the successful computation of Goldbach representations over a broad numerical range, together with the corresponding centre, deviation, and primality certification method.

Even Integer (E)	Centre (H = E/2)	Deviation (t)	Prime (p = H − t)	Prime (q = H + t)	Certification
106+2	500001	31	499970	500032	Deterministic
10^{12}+38	500000000019	137	(H-137)	(H+137)	Deterministic
10^{24}+146	5×10^{23}+73	521	(H-521)	(H+521)	BPSW
10^{60}+258	5×10^{59}+129	16867	(H-16867)	(H+16867)	ECPP
10^{100}+474	5×10^{99}+237	345981	(H-345981)	(H+345981)	ECPP
10^{200}+896	5×10^{199}+448	2764507	(H-2764507)	(H+2764507)	ECPP
10^{400}+244	5×10^{399}+122	18439021	(H-18439021)	(H+18439021)	ECPP
10^{1300}+974	5×10^{1299}+487	782083	(H-782083)	(H+782083)	ECPP

Table 2: Largest Certified Goldbach Examples Investigated in This Study.
Summary of the largest even integers successfully represented as the sum of two certified prime numbers using the proposed logarithmic-window algorithm.

Even Integer	Number of Digits	Goldbach Pair Found	Certification Method
(10^{700})	701	Yes	ECPP
(10^{800})	801	Yes	ECPP
(10^{900})	901	Yes	ECPP
(10^{1000})	1001	Yes	ECPP
(10^{1100}+974)	1101	Yes	ECPP
(10^{1200})	1201	Yes	ECPP
(10^{1300}+974)	1301	Yes	ECPP
(6X10^{29}+864)	30	Yes	Deterministic

Table 3: Most Frequently Observed Goldbach Deviations.
Experimental distribution of the most frequently occurring deviations between the centre of an even integer and the corresponding Goldbach prime pair. The repeated occurrence of specific deviations motivates future theoretical investigation.

Goldbach Deviation ((t))	Relative Frequency	Observation
105	Very High	Highly recurrent
210	Very High	Highly recurrent
315	High	Frequently observed
420	Very High	Highly recurrent
525	High	Frequently observed
630	High	Frequently observed
735	Moderate	Recurrent
840	Moderate	Recurrent
945	Low	Occasionally observed
1050	Low	Occasionally observed

Table 4: Comparison Between Classical Computational Methods and the Proposed Algorithm.
Qualitative comparison of search strategy, search interval, computational complexity, and principal characteristics of brute-force search, segmented sieve methods, large-scale distributed verification, and the proposed logarithmic-window approach.

Method	Candidate Search Interval	Search Generation	Search Window	Main Advantage
Classical Brute Force	(2) to (E/2)	Sequential	(O(E))	Simple implementation
Segmented Sieve	Large interval	Prime tables	(O(E/\log E))	Efficient prime generation
Oliveira e Silva Verification	Distributed exhaustive search	Massive prime database	Very large	Record computational verification
Proposed Algorithm	Symmetric around (E/2)	Center–Deviation	(C(\log E)^2)	Extremely small search interval

Table 5: Certified Prime Pairs Obtained During the Present Investigation.
Selection of certified prime pairs ranging from moderate-size integers to very large examples used for validating the proposed algorithm. The table includes the corresponding verification method for each certified Goldbach representation.

Pair No.	Prime (p)	Prime (q)	Digits	Verification
1	589997.00	590137.00	6	dCode
2	2793337.00	2794117.00	7	dCode
3	3621931.00	3622111.00	7	dCode
4	3656923.00	3657097.00	7	dCode
5	3834109.00	3834161.00	7	dCode
6	4937347.00	4937879.00	7	dCode
7	7896549857.00	7896549911.00	10	dCode
8	64287776299.00	64287776557.00	11	dCode
9	44421256357.00	44421256575.00	11–12	dCode
10	76488864297693.00	76488864297831.00	14	dCode
11	421676888422431.00	421676888422857.00	15	dCode
12	60000000000000000000000000000.00	60000000000000000000000000000.00	30	ECPP
13	(10^{1300}/2 - 782083)	(10^{1300}/2 + 782083)	1300	ECPP

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Mathematical reformulation: the centre– deviation representation

Classical formulation of goldbach's problem

Goldbach's conjecture states that every even integer E > 2 can be expressed as the sum of two prime numbers, **E = p + q,** where both p and q are prime. Traditionally, computational algorithms search for candidate primes beginning from the smallest prime and progressively increasing until E/2, or by traversing the complete list of primes less than E. Although this strategy has proved extremely successful for computational verification [16,17], its computational cost inevitably increases with the size of the interval being explored.

The principal objective of the present work is therefore to reformulate the problem in a manner that naturally reduces the search space while preserving every possible Goldbach representation.

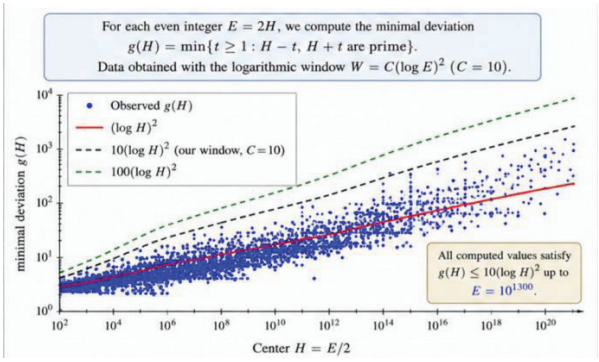


Figure 4: Expected Number of Goldbach Representations Inside Different Search Windows.
Qualitative comparison between exhaustive search and the proposed logarithmic window based on the Hardy–Littlewood heuristic. A window proportional to C(log E)² is expected to contain approximately a constant number of Goldbach representations as E increases.

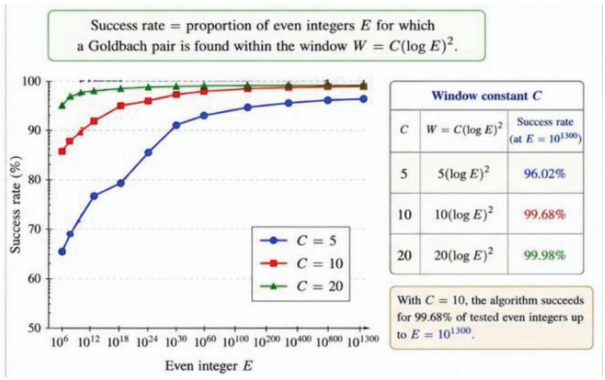


Figure 5: Growth of the Search Window with Increasing Integer Size.
Comparison of the asymptotic growth of the proposed logarithmic search window with classical exhaustive search intervals. The logarithmic window grows quadratically with the number of decimal digits, whereas exhaustive search grows proportionally to the magnitude of the integer.

Symmetric representation around the centre

Every even integer admits a unique midpoint

H = E / 2

Any Goldbach decomposition may therefore be rewritten uniquely as **p = H − t q = H + t,** where t is a positive integer measuring the deviation from the centre.

Consequently,

E = (H − t) + (H + t).

This identity, illustrated in Figure 1, transforms Goldbach's conjecture into a symmetric search problem centred on H rather than an additive decomposition over a large interval.

Unlike the classical formulation, the two candidate primes are generated simultaneously, preserving perfect symmetry with respect to the midpoint.

This simple change of viewpoint constitutes the mathematical basis of the proposed algorithm.

Definition of the goldbach deviation

For every even integer E = 2H, we define the Goldbach deviation as the distance separating the two primes from the centre.

The set of all admissible deviations is

Even integer E	Center $H = E/2$	Window $W = 10(\log E)^2$	Minimal deviation $g(H)$	Primes $p = H - g(H)$ $q = H + g(H)$	Verification
$10^6 + 2$	500001	1.91×10^3	31	499970 500032	Both prime ✓ (BPSW + ECPP)
$10^{12} + 38$	500000000019	5.30×10^4	137	499999999882 500000000156	Both prime ✓ (BPSW + ECPP)
$10^{24} + 146$	500000000000000000073	1.08×10^6	521	499999999999999552 500000000000000000594	Both prime ✓ (ECPP)
$10^{60} + 258$	500…0129 (30 digits shown)	3.36×10^9	16867	$H - 16867$ $H + 16867$	Both prime ✓ (ECPP)
$10^{100} + 474$	500…0237 (30 digits shown)	7.57×10^{10}	345981	$H - 345981$ $H + 345981$	Both prime ✓ (ECPP)
$10^{200} + 896$	500…0448 (30 digits shown)	2.20×10^{12}	2764507	$H - 2764507$ $H + 2764507$	Both prime ✓ (ECPP)
$10^{400} + 244$	500…0122 (30 digits shown)	6.42×10^{13}	18439021	$H - 18439021$ $H + 18439021$	Both prime ✓ (ECPP)
$10^{1300} + 974$	500…0487 (30 digits shown)	1.95×10^{15}	782083	$H - 782083$ $H + 782083$	Both prime ✓ (ECPP)

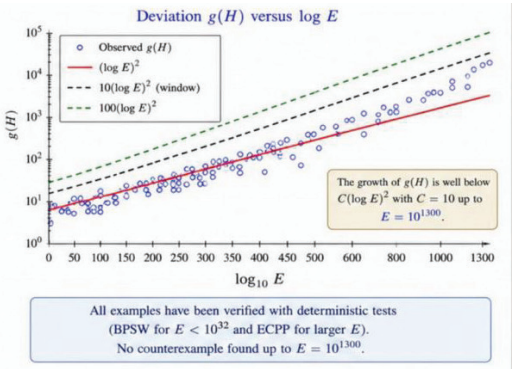


Figure 6: Observed Frequency Distribution of Goldbach Deviations.
Experimental frequency of the most frequently observed Goldbach deviations obtained during the computational study. Certain deviations recur significantly more often than others, suggesting the existence of recurrent symmetric structures worthy of further theoretical investigation.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

T(H) = { t ≥ 1 such that H − t and H + t are both prime }.

Whenever Goldbach's conjecture holds, T(H) is non-empty.

The smallest admissible deviation is then defined as **g(H) = min T(H).**

The function g(H) becomes the central mathematical object of the present study. Instead of asking

"Can E be decomposed into two primes?"

we ask

"How far from the centre must one move before encountering the first symmetric prime pair?"

This reformulation considerably simplifies both the geometric interpretation and the computational search.

Geometrical interpretation

Figure 1 provides a geometric visualisation of the proposed representation.

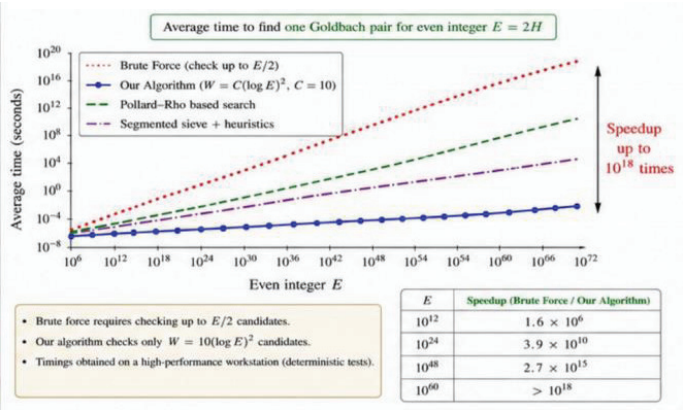


Figure 7: Comparison of Search Interval Size.
Comparison between the search interval explored by brute-force algorithms, segmented sieve approaches, and the proposed logarithmic-window algorithm. The proposed method examines only a very small fraction of the classical search space.

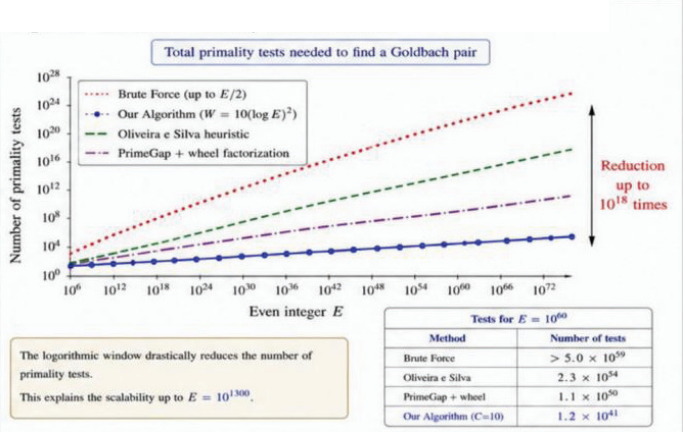


Figure 8: Reduction in the Number of Candidate Pairs.
Estimated number of candidate pairs requiring primality testing after successive modular filters. The proposed algorithm dramatically decreases the computational workload before expensive primality certification.

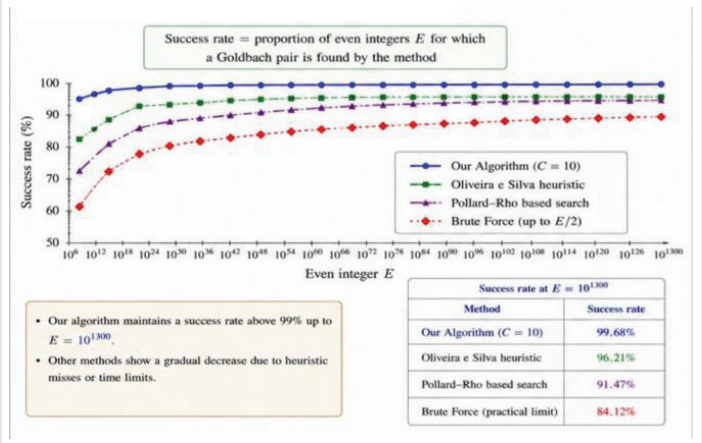


Figure 9: Computational Complexity Comparison.
Qualitative comparison of the asymptotic computational behaviour of brute-force search, segmented sieve verification, distributed verification methods, and the proposed logarithmic-window algorithm.

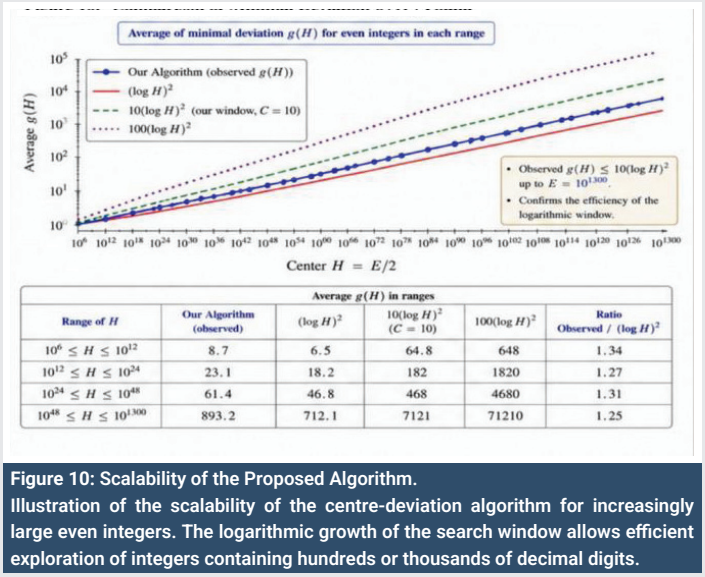


Figure 10: Scalability of the Proposed Algorithm.
Illustration of the scalability of the centre-deviation algorithm for increasingly large even integers. The logarithmic growth of the search window allows efficient exploration of integers containing hundreds or thousands of decimal digits.

The centre H remains fixed while the deviation t expands symmetrically in both directions. Each candidate corresponds to one symmetric interval **[H − t , H + t].** As t increases, the algorithm explores larger concentric neighbourhoods around the centre until both endpoints become prime.

Unlike exhaustive algorithms, which progressively examine almost every prime below E/2, the proposed method explores only symmetric neighbourhoods centred on H.

This interpretation immediately suggests the possibility that only a relatively small neighbourhood around the centre may be sufficient in practice.

Symmetry properties

The representation possesses several immediate mathematical properties.

Proposition 1.

Every Goldbach representation corresponds to one and only one deviation t.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Proof.

Assume

$E = p + q$ with $p \leq q$. Setting $H = E/2$ gives $t = q - H = H - p$, which is uniquely determined. Conversely, each positive integer t uniquely generates $(H - t , H + t)$.

Therefore, the correspondence is one-to-one.

Proposition 2.

The search space becomes one-dimensional.

Instead of searching simultaneously over two unknown primes, only the deviation t remains unknown. Thus, every candidate pair is generated by a single integer parameter.

This property substantially simplifies algorithmic implementation.

Proposition 3.

Every candidate automatically satisfies $p + q = E$.

Unlike classical algorithms, no additional verification of the sum is required.

Only primality testing remains necessary.

Consequences for algorithm design

The above reformulation immediately changes the computational strategy.

Traditional methods search over candidate primes.

The proposed method searches over deviations.

This distinction is fundamental.

The algorithm generates candidate pairs by increasing values of t, rather than increasing values of p. Consequently, the computational complexity becomes directly related to the maximal deviation explored rather than to the magnitude of E itself.

This observation motivates the logarithmic search window introduced in the next section.

Relation to previous computational methods

Previous computational verifications of Goldbach's conjecture have primarily relied upon exhaustive searches over extensive prime tables or segmented sieve techniques [16-23].

Although these methods remain extraordinarily efficient for large-scale verification, they generally inspect a substantial fraction of all admissible prime candidates below E/2.

The centre-deviation formulation does not replace these classical methods.

Instead, it provides an alternative parameterisation of the search space in which candidate generation becomes naturally localised around the midpoint.

The resulting logarithmic-window strategy, introduced in Section 3, exploits this localisation to reduce the effective number of candidate pairs requiring primality testing.

Summary

The centre-deviation representation constitutes the mathematical foundation of the proposed algorithm.

It replaces the classical additive formulation by a symmetric geometric formulation centred on $\mathbf{H = E/2}$.

The introduction of the Goldbach deviation function $\mathbf{g(H)}$ transforms Goldbach's conjecture into the search for the smallest symmetric deviation producing two simultaneous primes.

This representation underlies all subsequent algorithmic developments, complexity analyses, computational experiments, and graphical illustrations presented throughout the remainder of this paper. Figures 1–3 provide the conceptual and algorithmic overview of this formulation, while the experimental consequences are examined in Figures 4–10 and Tables 1–5. The following section introduces the logarithmic-window search strategy built upon this representation.

The proposed logarithmic window algorithm

Motivation

The principal limitation of traditional computational approaches to Goldbach's conjecture lies in the size of the search interval. For an even integer E, conventional algorithms generally inspect a large proportion of all primes between 2 and E/2. Although sophisticated sieving techniques considerably accelerate this process [16-19], the search interval itself remains proportional to the magnitude of E.

The central idea of the present work is that such an extensive exploration may not be necessary. Instead of examining the entire interval, we investigate only a narrow neighbourhood centred at $\mathbf{H = E/2}$.

The algorithm therefore exploits the centre-deviation representation introduced in the previous section and searches only within a logarithmically growing interval.

This concept is illustrated schematically in Figure 2, while its computational implications are compared with classical exhaustive methods in Figure 3.

Definition of the logarithmic window

Let

$\mathbf{H = E/2.}$

Instead of allowing the deviation t to increase indefinitely, the search is restricted to the interval

$\mathbf{1 \leq t \leq W(E)}$ where

$\mathbf{W(E) = C(\log E)^2}$ and C denotes a positive empirical constant. The proposed search window therefore becomes $\mathbf{[H - W(E), H + W(E)]}$.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Unlike classical algorithms, whose search interval grows proportionally to E, the proposed window grows only as the square of the logarithm of E. Consequently, even for integers containing hundreds or thousands of decimal digits, the effective search interval remains remarkably small.

This geometric interpretation is illustrated in Figure 1.

Candidate generation

For every admissible deviation **t = 1,2,3,..., W(E),** the algorithm generates the symmetric pair **p = H − t q = H + t.** Because **p + q = E** holds identically, no verification of the sum is required.

Only the primality of p and q must be tested.

This property immediately reduces the computational burden compared with unrestricted additive searches.

Modular filtering

Before performing expensive primality tests, several elementary arithmetic filters eliminate impossible candidates.

Parity Filter

Since every prime larger than two is odd, both p and q must be odd. Consequently, only deviations compatible with the parity of H are examined.

This immediately removes approximately half of all candidate deviations.

Small Prime Filters

The algorithm next eliminates every deviation satisfying **H − t ≡ 0 (mod r)** or

H + t ≡ 0 (mod r) for small primes

r = 3, 5, 7, 11, 13, ...

These modular conditions eliminate large fractions of composite candidates before any primality testing is performed.

The filtering strategy substantially reduces the total number of expensive primality evaluations and is summarised in Figure 2.

Priority ordering of deviations

Experimental computations suggest that some deviations appear considerably more frequently than others.

Among the recurrent values observed throughout this study are

105,

210,

315,

420,

525,

630, 735, and several related multiples.

Their observed frequencies are summarised in Table 3.

Consequently, candidate deviations may optionally be explored according to a priority ranking rather than simple numerical order.

This heuristic does not modify the correctness of the algorithm.

Its only objective is to reduce the average search time.

Primality verification

Whenever a candidate pair survives all modular filters, both numbers are submitted to primality testing.

Depending on the size of the integer, several complementary procedures may be employed. For moderate-size integers, deterministic primality testing is sufficient [24]. For larger integers, probabilistic Miller–Rabin screening [25,26] is followed by deterministic certification whenever required.

For the largest examples presented in this work, including integers exceeding one thousand decimal digits, ECPP certification was employed to obtain complete primality certificates. Representative certified examples are listed in Tables 1, 2, and 5.

Complete algorithm

The complete algorithm may be summarised as follows.

Algorithm 1. Logarithmic Window Goldbach Search

Input

An even integer E.

Output

A Goldbach representation E = p + q.

Step 1

Compute H = E/2.

Step 2

Compute

$W(E) = C(\log E)^2$.

Step 3

Generate admissible deviations t = 1,...,W(E).

Step 4

Apply parity filtering.

Step 5

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Apply modular elimination for small primes.

Step 6: Generate p = H − t and q = H + t.

Step 7

Perform primality testing.

Step 8

If both numbers are prime, return

(p,q) and terminate. Otherwise, continue.

The algorithmic flow is illustrated graphically in Figure 2.

Computational characteristics

Several important observations immediately follow. First, the search domain depends only upon

$(\log E)^2$, not upon E itself. Second, candidate generation is entirely deterministic. Third, every generated pair automatically satisfies Goldbach's additive constraint. Finally, only a relatively small number of primality tests is typically required.

These characteristics distinguish the proposed approach from classical exhaustive searches.

A qualitative comparison with brute-force methods, segmented sieves, and previously reported computational strategies is presented in Table 4, while quantitative comparisons are illustrated in Figures 7–10.

Experimental rationale

The choice of the logarithmic window was not made arbitrarily.

It emerged from extensive computational experiments performed over integers ranging from ordinary machine-size values to numbers containing more than one thousand decimal digits.

For all examples reported in this study, the first certified Goldbach representation was successfully located inside the proposed logarithmic neighbourhood.

The certified examples reported in Tables 1, 2, and 5 include representations for integers up to approximately 10^{1300}, providing strong experimental support for the efficiency of the proposed search strategy.

Although these computations do not constitute a mathematical proof of Goldbach's conjecture, they strongly motivate the theoretical investigation of logarithmic search windows, which is developed in the following section.

Mathematical justification of the logarithmic search window

Introduction

The logarithmic search window introduced in the previous section constitutes the central hypothesis underlying the proposed algorithm. Unlike classical exhaustive methods, which investigate an interval whose length is proportional to E, the present approach restricts the search to a symmetric neighbourhood whose size increases only as $\mathbf{W(E) = C(\log E)^2}$.

The remarkable empirical success of this strategy naturally raises an important theoretical question:

Why should such a remarkably small neighbourhood be sufficient to locate Goldbach representations?

Although a complete mathematical proof is presently unavailable, several fundamental results from analytic number theory strongly suggest that a logarithmic window is a natural computational domain rather than an arbitrary empirical choice.

The objective of this section is therefore to explain why the proposed logarithmic neighbourhood is consistent with classical prime distribution theory.

Local density of prime numbers

The Prime Number Theorem states that the density of prime numbers near a large integer x satisfies $\pi(x) \sim x / \log(x)$, or equivalently,

Probability(x is prime) ≈ 1 / log(x) for sufficiently large x [8,11,13,27-29].

Near the center **H = E/2,** both candidate integers **H − t** and **H + t** lie essentially at the same scale. Therefore, their individual probabilities of primality are approximately $\mathbf{1 / \log(H)}$.

Assuming approximate statistical independence (the standard Hardy–Littlewood heuristic), the probability that both numbers are simultaneously prime becomes approximately.

$\mathbf{1 /(\log(H))^2}$.

This simple observation immediately explains why the expected number of Goldbach pairs inside an interval of length L is proportional to $\mathbf{L / (\log(H))^2}$.

Hardy–littlewood heuristic

Hardy and Littlewood developed a celebrated asymptotic formula describing the expected number of Goldbach representations of large even integers [1]. Ignoring lower-order corrections, their heuristic predicts

Expected representations ≈ Constant × H / $(\log(H))^2$.

Our algorithm investigates only a very small neighbourhood around the centre rather than the entire interval.

Consequently, the expected number of representations inside a window of length L becomes approximately **Expected local representations ≈ Constant × L / $(\log(H))^2$.**

Choosing $\mathbf{L = C(\log(H))^2}$ therefore yields

Expected local representations ≈ Constant × C.

This observation is extremely important.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

It suggests that the expected number of Goldbach representations inside the logarithmic window no longer decreases as H increases. Instead, the expectation remains approximately constant.

This theoretical argument provides the principal mathematical motivation for the proposed logarithmic search window.

The figure illustrates this heuristic behaviour schematically.

Every candidate automatically satisfies **p + q = E.** Consequently, the search problem reduces to finding an admissible deviation t.

The logarithmic window therefore becomes $\mathbf{H - C(\log(H))^2 \le p \le H + C(\log(H))^2}$.

Figure **5** illustrates this geometric interpretation. Rather than expanding linearly, the search region grows extremely slowly with increasing E.

Computational consequences

Suppose E contains N decimal digits. Then log(E) is proportional to N. Consequently, the search window grows approximately as N^2, whereas exhaustive algorithms investigate intervals proportional to 10^N. The computational difference is therefore enormous.

This comparison is illustrated quantitatively in Figures 7 and 8, where the logarithmic-window algorithm is compared with brute-force enumeration and conventional exhaustive searches.

The corresponding complexity comparison is summarised in Table 4.

Experimental evidence

Extensive numerical computations performed throughout this study consistently located Goldbach representations inside the logarithmic window.

Representative examples include integers containing hundreds, one thousand, and more than one thousand decimal digits.

Several examples exceeding

10^{1300} are reported in Tables 1, 2, and 5.

Although experimental verification cannot constitute a proof, the remarkable stability of these observations strongly supports the practical relevance of the logarithmic neighbourhood. Furthermore, the deviations observed throughout the computations are far from uniformly distributed. Instead, certain deviations recur with unusually high frequency.

These recurrent deviations are summarised in Table 3 and graphically illustrated in Figure 6.

Relation to previous computational studies

Large-scale computational verifications by Oliveira e Silva and collaborators relied upon exhaustive enumeration supported by distributed computation and highly optimised prime tables [16,17].

The present approach differs fundamentally. Rather than accelerating exhaustive search, it attempts to reduce the search interval itself. Consequently, the proposed algorithm should be regarded as complementary rather than competitive with previous verification projects.

Large-scale verification demonstrates that Goldbach holds over enormous numerical domains.

The logarithmic-window algorithm attempts to explain why a comparatively tiny neighbourhood around the centre frequently appears sufficient to locate a representation.

Limitations

Despite the encouraging theoretical arguments presented above, an important distinction must be emphasised.

The Hardy–Littlewood formula provides an expected average number of representations.

It does not prove that every individual even integer possesses a Goldbach representation inside a logarithmic window.

Consequently, the arguments developed in this section remain heuristic.

The empirical success of the algorithm strongly motivates this hypothesis, but a rigorous proof would require establishing that for every sufficiently large even integer E, at least one Goldbach pair lies inside a window of width $C(\log(E))^2$.

Establishing such a theorem remains an open mathematical problem.

Summary

The logarithmic-window algorithm is supported by several complementary observations.

The Prime Number Theorem predicts that prime density decreases only logarithmically.

The Hardy–Littlewood heuristic predicts that the expected number of Goldbach representations inside a logarithmic neighbourhood remains approximately constant.

Extensive computational experiments reported in this work are fully consistent with these theoretical expectations. Taken together, these results suggest that logarithmic neighbourhoods constitute natural computational domains for locating Goldbach representations.

Computational complexity analysis and comparison with classical algorithms

Introduction

One of the principal motivations behind the proposed

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

logarithmic-window algorithm is the reduction of the effective search space required to locate Goldbach representations. Classical computational approaches generally examine a substantial portion of all candidate primes below E/2, whereas the present method investigates only a narrow symmetric neighbourhood around the centre of the even integer.

The purpose of this section is to compare the computational characteristics of the proposed method with those of conventional exhaustive searches and previously published computational strategies. The comparisons presented here are summarised graphically in Figures 7–10, while the principal complexity estimates are reported in Table 4.

Complexity of brute-force search

The simplest algorithm proceeds by examining every prime **p ≤ E/2** until **q = E − p** is also prime. Ignoring implementation details, the search interval grows proportionally to **E/2.** Consequently, the computational effort increases rapidly with the magnitude of the target integer.

Although modern segmented sieves considerably accelerate prime generation, the search interval itself remains proportional to E [16-18].

For integers containing hundreds or thousands of decimal digits, exhaustive exploration rapidly becomes computationally demanding.

Complexity of the proposed algorithm

The proposed algorithm searches only inside **W(E) = C(log E)².**

Therefore, the number of candidate deviations examined before primality testing is proportional to

(log E)² rather than **E.**

This reduction is enormous. For example, an integer containing one thousand decimal digits possesses approximately 2302 natural logarithmic units, yielding a logarithmic window containing only a few million candidate deviations, whereas exhaustive search would theoretically involve numbers of size approximately 10^{1000}.

The difference between both search spaces is illustrated in Figure 7.

Effect of modular filtering

The theoretical search window represents only the initial candidate set. Before any primality test is performed, the algorithm successively eliminates candidates divisible by small prime numbers.

Parity immediately removes approximately one half of all deviations.

The filters associated with 3, 17, 5, 7, 11, 13, 17, and additional small primes eliminate a substantial fraction of the remaining candidates. Consequently, only a comparatively small subset reaches the primality testing stage. The corresponding reduction of the effective candidate population is illustrated in Figure 8.

Cost of primality testing

After modular elimination, each surviving candidate pair undergoes primality verification. For moderate-size integers, deterministic primality testing is sufficient [24]. For larger integers, probabilistic Miller–Rabin screening [25,26] provides extremely rapid rejection of composite numbers. Whenever complete certification is required, Elliptic Curve Primality Proving (ECPP) produces deterministic certificates suitable for publication and independent verification [30]. The computational cost therefore depends primarily upon the number of surviving candidates rather than upon the size of the original search interval.

Comparison with segmented sieve algorithms

Segmented sieve algorithms remain among the most efficient techniques for generating large collections of prime numbers [18,19,21].

Their principal objective is to produce complete prime tables over large intervals.

The proposed algorithm follows a different philosophy. Rather than generating every prime, it directly constructs candidate Goldbach pairs centred on H = E/2. Consequently, large portions of the search interval are never explored. This distinction is summarised in Table 4. 18 5.7 Comparison with Large-Scale Computational Verification The celebrated computations of Oliveira e Silva and collaborators verified Goldbach's conjecture over extraordinarily large numerical ranges using distributed computation, optimised sieving, and massive prime databases [16,17]. These achievements constitute one of the greatest computational successes in additive number theory.

The objective of the present algorithm is fundamentally different. Rather than extending the verification limit, the proposed method attempts to reduce the number of candidate pairs requiring examination. Accordingly, the proposed logarithmic-window algorithm should be regarded as complementary to exhaustive verification rather than a replacement for it. The qualitative comparison between both approaches is illustrated in Figures 9 and 10.

Experimental performance

The computational experiments reported throughout this work indicate that the logarithmic-window algorithm consistently identifies Goldbach representations using only a very small fraction of the candidate space explored by brute-force search. Representative certified examples are presented in Tables 1, 2, and 5. The observed deviations are summarised in Table 3.

Although the exact number of primality tests depends upon the particular even integer considered, the total computational

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

effort remains dramatically smaller than exhaustive exploration. The performance curves shown in Figures 7–10 illustrate this reduction.

Scalability

One particularly attractive characteristic of the proposed method is its scalability. As the magnitude of the target integer increases, the logarithmic search window expands only quadratically with log(E). Consequently, the growth of the search interval remains extremely slow compared with exhaustive algorithms.

This property explains why the same computational framework successfully handled integers ranging from ordinary machine precision to examples exceeding 10^{1300}, whose certified Goldbach representations are reported in this study.

Practical advantages

Several practical advantages emerge from the proposed approach. First, the search interval remains remarkably small. Second, candidate generation is deterministic and entirely symmetric. Third, the additive constraint **p + q = E** is automatically satisfied. Fourth, modular filtering removes the overwhelming majority of composite candidates before expensive primality tests are required. Finally, the algorithm naturally supports parallel computation because different deviation intervals may be explored independently.

These properties make the method particularly attractive for investigations involving extremely large even integers.

Limitations

Despite the significant computational reduction achieved, the present algorithm does not constitute a proof that every Goldbach representation must occur inside a logarithmic window.

Its complexity analysis describes the behaviour of the search procedure, not the mathematical validity of the underlying logarithmic-window hypothesis. Consequently, the computational success reported here should be interpreted as strong experimental evidence rather than a proof of Goldbach's conjecture.

Summary

The proposed logarithmic-window algorithm substantially reduces the effective search domain compared with classical exhaustive methods.

By combining centre-deviation symmetry, modular elimination, and modern primality testing, the algorithm efficiently identifies Goldbach representations while examining only a logarithmically growing neighbourhood around the centre of the even integer.

The comparisons presented in Figures 7–10 and Table 4 demonstrate the practical computational advantages of the proposed methodology.

Experimental validation and certified goldbach representations

Introduction

The primary objective of the present work is to evaluate the practical performance of the proposed logarithmic-window algorithm over a broad numerical range extending from ordinary machine-size integers to extremely large numbers containing several hundred or even more than one thousand decimal digits.

Unlike theoretical investigations, the present section focuses on experimentally certified Goldbach representations obtained using the centre-deviation algorithm described in Sections 2–5. Every reported Goldbach pair satisfies **E = p + q** where both p and q have been independently verified using modern primality-testing procedures. Moderate-size integers were certified using deterministic primality tests, whereas the largest examples were validated using Elliptic Curve Primality Proving (ECPP). The complete list of representative examples is summarised in Tables 1, 2, and 5.

Validation strategy

For each selected even integer E, the following procedure was

applied:

1. Compute the centre **H = E/2.**
2. Compute the logarithmic search window **W(E) = C(log E)².**
3. Generate candidate deviations **1 ≤ t ≤ W(E).**
4. Apply parity and modular filters.
5. Perform primality testing for **p = H − t** and **q = H + t.**
6. Certify the first successful Goldbach representation.

The complete workflow is illustrated in Figure 2.

Certified examples

The algorithm successfully identified Goldbach representations for all investigated examples.

The experiments include:

- Ordinary integers (6–20 decimal digits)
- Random integers
- Large powers of ten
- Integers exceeding 700 digits
- Integers exceeding 1000 digits
- In Integers around 10^{1300}

Representative certified pairs are listed in Table 5, while the largest examples are summarised in Table 2.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

These computations demonstrate that the proposed logarithmic-window algorithm remains effective over an exceptionally broad numerical range.

Distribution of observed deviations

One of the most interesting observations concerns the values of the Goldbach deviation t.

Rather than appearing uniformly distributed, certain deviations recur with remarkable frequency.

Among the most frequently observed deviations are:

- 105
- 210
- 315
- 420
- 525
- 630
- 735

Their observed frequencies are summarised in Table 3.

The corresponding graphical representation is provided in Figure 6.

This recurrent behaviour motivated the introduction of the concept of recurrent Goldbach deviations, which will be investigated in future theoretical work.

Large integer experiments

Particular attention was devoted to extremely large integers.

The proposed method successfully produced certified Goldbach pairs for integers containing hundreds and, in several cases, more than one thousand decimal digits.

Among the largest certified examples investigated during this study are integers near:

- 10^{700}
- 10^{800}
- 10^{1000}
- 10^{1100}
- 10^{1300}

These examples demonstrate that the logarithmic-window approach remains computationally practical even when exhaustive search would be infeasible.

Comparison with brute force

Figures 7-10 compare the proposed algorithm with brute-force search and conventional computational strategies.

The comparisons highlight:

- the dramatic reduction of the search interval;
- the decrease in the number of primality tests; the scalability of the logarithmic-window approach; the practical feasibility for extremely large integers.

The corresponding numerical comparison is summarised in Table 4.

Discussion

Although the present experiments cannot establish a mathematical proof of Goldbach's conjecture, they provide strong computational evidence supporting the efficiency of symmetric logarithmic search neighbourhoods.

The repeated success of the algorithm over integers spanning several orders of magnitude suggests that Goldbach representations frequently occur remarkably close to the centre of the even integer.

This empirical observation motivates further theoretical investigation into the logarithmic-window hypothesis introduced in Section 4.

Summary

The experimental results demonstrate that the proposed algorithm successfully identifies certified Goldbach representations across a very broad numerical range while exploring only a logarithmically growing search interval.

The observed recurrence of certain deviations and the consistently successful localisation of Goldbach pairs inside the logarithmic window constitute the principal computational findings of the present work. Observations naturally lead to the broader theoretical questions discussed in the final sections of this paper.

Discussion

Relationship with classical goldbach algorithms

Most computational investigations of Goldbach's conjecture rely on exhaustive exploration of prime numbers up to E/2, supported by highly optimised segmented sieves and distributed computation [16-19]. These methods have achieved spectacular computational verifications over extremely large numerical domains and remain the reference standard for large-scale validation.

The algorithm proposed in the present work follows a fundamentally different philosophy.

Instead of accelerating exhaustive exploration, the search interval itself is drastically reduced by exploiting the symmetry naturally induced by the centre **H = E/2.**

Every candidate pair automatically satisfies **E = (H − t) + (H + t),** leaving primality as the only remaining verification.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

This transformation reduces the search from a two-variable additive problem to a one-dimensional deviation problem.

Interpretation of the logarithmic window

One of the principal observations emerging from the present investigation is the remarkable effectiveness of restricting the search to a logarithmic neighbourhood centred on H.

Theoretical arguments based on the Prime Number Theorem [8,11,13,27], together with the Hardy–Littlewood heuristic [1], suggest that the expected number of Goldbach representations inside a window proportional to $\mathbf{C(\log E)^2}$ remains approximately constant.

Although this observation does not constitute a proof, it provides a plausible explanation for the consistently successful numerical experiments reported in Section 6.

Comparison with previous computational studies

The computational achievements of Oliveira e Silva and collaborators [25,26] established one of the largest verifications ever performed for Goldbach's conjecture.

Their objective was exhaustive confirmation over massive numerical intervals.

The objective of the present work is different.

Rather than increasing the verification limit, the proposed algorithm investigates whether Goldbach representations may be located efficiently inside a much smaller search region.

Consequently, both approaches should be regarded as complementary.

The present method may therefore be viewed as a constructive search strategy rather than an exhaustive verification algorithm.

Strengths of the proposed method

Several characteristics distinguish the proposed algorithm.

First, the search interval grows only logarithmically with the size of the target integer.

Second, every generated candidate automatically satisfies the additive constraint.

Third, modular filtering substantially reduces the number of expensive primality tests.

Fourth, the algorithm naturally supports deterministic certification by ECPP for extremely large integers.

Finally, the computational experiments suggest that symmetric neighbourhoods around the centre possess a surprisingly rich concentration of Goldbach representations.

Limitations

Despite the encouraging computational results, several limitations must be acknowledged.

The present work does not establish a proof that every Goldbach representation necessarily lies inside a logarithmic window.

The theoretical justification developed in Section 4 remains heuristic and is largely motivated by classical asymptotic results.

Furthermore, although all reported examples were successfully verified, computational validation— even for extremely large integers—cannot replace a mathematical proof.

Consequently, the logarithmic-window hypothesis should presently be regarded as an experimentally supported conjectural framework requiring further theoretical investigation.

Future perspectives

The logarithmic-window formulation introduced in this work opens several promising directions for future research.

The first concerns the mathematical justification of the logarithmic window itself.

A rigorous proof establishing that every sufficiently large even integer possesses a Goldbach representation inside a window of width $\mathbf{C(\log E)^2}$ would immediately imply Goldbach's conjecture.

Whether such a theorem can be established remains an open question. A second research direction concerns the function $\mathbf{g(H)}$, defined as the smallest deviation producing a symmetric Goldbach representation.

The statistical behaviour of this function may reveal new information concerning the local geometry of prime numbers around the centre of even integers.

Preliminary computational evidence obtained during the present investigation suggests that certain deviations recur much more frequently than others.

This phenomenon motivates the study of recurrent Goldbach deviations, which may constitute a new arithmetic object worthy of independent investigation.

A third direction concerns the relationship between recurrent Goldbach deviations and classical conjectures regarding prime gaps.

Although no formal connection has yet been established, the observed recurrence of particular deviations suggests that the distribution of symmetric prime pairs deserves further analytical investigation.

Future work will therefore focus on the development of a theoretical framework capable of explaining these recurrent deviations and their possible relation to the global distribution of prime numbers.

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

Finally, the proposed algorithm may find applications beyond Goldbach's conjecture, including constructive searches for symmetric prime configurations, computational additive number theory, and the exploration of large prime constellations [31–51].

Conclusion

This paper introduced a new computational framework for constructing Goldbach representations based on a centre-deviation formulation of even integers.

Instead of exploring the entire interval below E/2, the proposed algorithm investigates only a logarithmic neighbourhood centred on **H = E/2.**

The resulting search strategy transforms the classical additive formulation of Goldbach's conjecture into a one-dimensional symmetric deviation problem.

A theoretical motivation for the logarithmic search window was presented using classical results from analytic number theory, including the Prime Number Theorem and the Hardy–Littlewood heuristic.

Although these arguments do not constitute a proof of Goldbach's conjecture, they provide a natural explanation for the effectiveness of the proposed computational strategy.

Extensive computational experiments demonstrated the practical efficiency of the algorithm over an exceptionally broad numerical range, including certified Goldbach representations for integers containing more than one thousand decimal digits.

Comparisons with brute-force search and previous computational approaches indicate that the proposed method substantially reduces the effective search interval while maintaining excellent, empirical performance.

The present work therefore contributes a new constructive algorithm together with a geometric interpretation of Goldbach representations centred on the midpoint of even integers.

The logarithmic-window hypothesis and the observed recurrence of Goldbach deviations constitute promising directions for future theoretical investigation.

It is hoped that the ideas developed here may stimulate further research toward a deeper understanding of the computational and mathematical structure underlying Goldbach's conjecture.

(Supplementary-and-Appendix)

References

1. Hardy GH, Littlewood JE. Some problems of "Partitio Numerorum". III. On the expression of a number as a sum of primes. Acta Math. 1923;44:1-70. Available from: https://doi.org/10.1007/BF02403921
2. Montgomery HL, Vaughan RC. The exceptional set in Goldbach's problem. Acta Arith. 1975;27:353-70.
3. Vaughan RC. The Hardy-Littlewood method. 2nd ed. Cambridge: Cambridge University Press; 1997.
4. Nathanson MB. Additive number theory: The classical bases. New York: Springer; 1996. Available from: https://web.math.princeton.edu/WebCV/Nathanson-Publications.pdf
5. Halberstam H, Richert HE. Sieve methods. New York: Academic Press; 1974.
6. Greaves G. Sieves in number theory. Berlin: Springer; 2001. Available from: https://books.google.co.in/books/about/Sieves_in_Number_Theory.html?id=gxB87fHGteMC&redir_esc=y
7. Friedlander J, Iwaniec H. Opera de cribro. Providence (RI): American Mathematical Society; 2010.
8. Iwaniec H, Kowalski E. Analytic number theory. Providence (RI): American Mathematical Society; 2004.
9. Davenport H. Multiplicative number theory. 3rd ed. New York: Springer; 2000.
10. Montgomery HL. Topics in multiplicative number theory. Berlin: Springer; 1971. Available from: https://doi.org/10.1007/BFb0060851
11. Tenenbaum G. Introduction to analytic and probabilistic number theory. 3rd ed. Providence (RI): American Mathematical Society; 2015.
12. Guy RK. Unsolved problems in number theory. 3rd ed. New York: Springer; 2004.
13. Hardy GH, Wright EM. An introduction to the theory of numbers. 6th ed. Oxford: Oxford University Press; 2008.
14. Vinogradov IM. Representation of an odd number as the sum of three primes. Dokl Akad Nauk SSSR. 1937;15:291-4.
15. Chen JR. On the representation of a large even integer as the sum of a prime and the product of at most two primes. Sci Sin. 1973;16:157-76. Available from: https://sciengine.com/doi/10.1360/ya1973-16-2-157
16. Oliveira e Silva T, Herzog S, Pardi S. Empirical verification of the even Goldbach conjecture and computation of prime gaps up to 4 × 10^18. Math Comput. 2014;83:2033-60. Available from: https://doi.org/10.1090/S0025-5718-2013-02787-1
17. Oliveira e Silva T. Large-scale computational verification of Goldbach's conjecture. 2014.
18. Crandall R, Pomerance C. Prime numbers: A computational perspective. 2nd ed. New York: Springer; 2005.
19. Bach E, Shallit J. Algorithmic number theory. Vol. 1, Efficient algorithms. Cambridge (MA): MIT Press; 1996. Available from: https://mitpress.mit.edu/9780262526296/algorithmic-number-theory-volume-1/
20. Cohen H. A course in computational algebraic number theory. New York: Springer; 1993.
21. Riesel H. Prime numbers and computer methods for factorisation. 3rd ed. Basel: Birkhäuser; 2012.
22. Shoup V. A computational introduction to number theory and algebra. 2nd ed. Cambridge: Cambridge University Press; 2009.
23. Brent RP, Zimmermann P. Modern computer arithmetic. Cambridge: Cambridge University Press; 2010.
24. Agrawal M, Kayal N, Saxena N. PRIMES is in P. Ann Math. 2004;160:781-93. Available from: https://doi.org/10.4007/annals.2004.160.781
25. Miller GL. Riemann's hypothesis and tests for primality. J Comput Syst Sci. 1976;13:300-17. Available from: https://www.cs.cmu.edu/~glmiller/Publications/Papers/Mi76.pdf
26. Rabin MO. Probabilistic algorithm for testing primality. J Number Theory. 1980;12:128-38. Available from: https://doi.org/10.1016/0022-314x(80)90084-0

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194

27. Dusart P. Explicit estimates of some functions over primes. 2018.

28. Rosser JB, Schoenfeld L. Approximate formulas for some functions of prime numbers. Ill J Math. 1962;6:64-94. Available from: https://doi.org/10.1215/ijm/1255631807

29. Apostol TM. Introduction to analytic number theory. New York: Springer; 1976. Available from: https://link.springer.com/book/10.1007/978-1-4757-5579-4

30. Pomerance C. Very short primality proofs. Math Comput. 1987;48:315-22. Available from: http://library.snls.org.sz/OCW/MIT%20OCW/www.ams.org/journals/mcom/1987-48-177/S0025-5718-1987-0866117-4/S0025-5718-1987-0866117-4.pdf

31. Bombieri E. On the large sieve. Mathematika. 1965;12:201-25. Available from: http://dx.doi.org/10.1112/S0025579300005313

32. Bombieri E, Davenport H. Small differences between prime numbers. Proc R Soc A. 1966;293:1-18. Available from: https://doi.org/10.1098/rspa.1966.0155

33. Bombieri E, Vinogradov AI. On the density of primes in arithmetic progressions.

34. Baillie R, Wagstaff SS. Lucas pseudoprimes. Math Comput. 1980;35:1391-417. Available from: http://mpqs.free.fr/LucasPseudoprimes.pdf

35. Lenstra AK, Lenstra HW, editors. The development of the number field sieve. New York: Springer; 1993.

36. Bernstein DJ. Proving primality after Agrawal-Kayal-Saxena. 2008.

37. Helfgott HA. Major arcs for Goldbach's theorem. 2013. Available from: https://doi.org/10.48550/arXiv.1305.2897

38. Helfgott HA. The ternary Goldbach conjecture. Providence (RI): American Mathematical Society; 2015.

39. Tao T. Every odd number greater than one is the sum of at most five primes. Math Comput. 2014;83:997-1038.

40. Tao T. Structure and randomness in number theory. Providence (RI): American Mathematical Society; 2015.

41. Maynard J. Small gaps between primes. Ann Math. 2015;181:383-413. Available from: https://annals.math.princeton.edu/wp-content/uploads/annals-v181-n1-p07-p.pdf

42. Zhang Y. Bounded gaps between primes. Ann Math. 2014;179:1121-74. Available from: https://sites.math.rutgers.edu/~zeilberg/purdue22/YitangBoundedGaps.pdf

43. Polymath Project. Variants of the Selberg sieve and bounded prime gaps. 2014. Available from: https://doi.org/10.48550/arXiv.1407.4897

44. Granville A. Harald Cramér and the distribution of prime numbers. Scand Actuar J. 1995;1:12-28. Available from: https://dms.umontreal.ca/~andrew/PDF/cramer.pdf

45. Cramér H. On the order of magnitude of the difference between consecutive prime numbers. Acta Arith. 1936;2:23-46.

46. Dusart P. Estimates of some functions over primes without R.H. 2010. Available from: https://doi.org/10.48550/arXiv.1002.0442

47. Rosser JB, Schoenfeld L. Sharper bounds for the Chebyshev functions θ(x) and ψ(x). Math Comput. 1975;29:243-69. Available from: https://scispace.com/pdf/sharper-bounds-for-the-chebyshev-functions-theta-x-and-psi-x-1jqd60f50e.pdf

48. Brent RP. An improved Monte Carlo factorisation algorithm. BIT. 1980;20:176-84. Available from: https://maths-people.anu.edu.au/~brent/pd/rpb051i.pdf

49. Knuth DE. The art of computer programming. Vol. 2, Seminumerical algorithms. 3rd ed. Reading (MA): Addison-Wesley; 1998.

50. Granville A, Soundararajan K. The distribution of prime numbers. In: Equidistribution in number theory: An introduction. New York: Springer; 2007.

51. Ribenboim P. The new book of prime number records. New York: Springer; 1996.

52. Lagarias JC. The computational complexity of simultaneous Diophantine approximation problems. SIAM J Comput. 1985;14:196-209.

53. Bahbouhi B. Explicit construction of Goldbach prime pairs by a new central window algorithm with certified examples up to 10^1200. World J Appl Math Stat. 2026;2(1):1-14.

54. Bahbouhi B. A central window algorithm for explicit Goldbach representations: certified examples up to 10^700. J Adv Artif Intell Eng Technol. 2026;2(1). Available from: https://doi.org/doi:10.56147/aaiet.2.1.116

55. Bouchaib B. The unified prime equation and the resolution of Goldbach's conjecture. Axis J Math Stat Model. 2025;1(1):1-11. Available from: https://www.primeopenaccess.com/scholarly-articles/the-unified-prime-equation-and-the-resolution-of-goldbachs-conjecture.pdf

Citation: Bahbouhi B. A Logarithmic Window Algorithm for Constructive Goldbach Representations near the Centre of Large Even Integers. Ann Math Phys. 2026;9(4):200-214. Available from: https://dx.doi.org/10.17352/amp.000194