

Bbynalog Logic : Prime Factorization & Diophantine Equations

by Alli-Smith Laryairn Neutrialia

ABSTRACT

Factorization of a co-prime number into its component prime numbers, is considered to be a problem that does not have a polynomial time solution, the current optimal algorithm appears to be Shor's quantum computing algorithm. Finding the right integer values for some diophantine equations is also considered to be np-hard, In this paper we consider a generalized polynomial time method for both prime factorization and diophantine integer resolution; (Alli-Smith Signal Method), that can be easily implemented on classical computers.

OUTLINE

- 1). Introduction
- 2). Alli-Smith Signal Factorization Method (A.S.S.F.M)
- 3). Riemann Hypothesis
- 4). Diophantine Equations
- 5). Alli-Smith Signal Diophantine Method (A.S.S.D.M)

- 6). Conclusion
- 7). References

INTRODUCTION

Consider an integer co-prime number d that is composed of two integer prime numbers that both have very large number of digits; n & m , finding the prime numbers n & m is considered to be a non-polynomial time problem with current classical techniques.

$$d = n * m$$

The generally agreed upon algorithm is to test if d/x has a solution y ;

$$d/x = y$$

Where;

x is an integer number from 1 to $\text{Floor}[\text{Sqrt}[d]]$

& for every x ; y is an integer solution: $x \leq \text{Floor}[\text{Sqrt } d] < y$

If for any x from the interval above there is an integer solution y then x & y are considered to be the co-prime numbers of d .

Let B : Prime Factorization Equation of form $(x * y) - c = 1$

$$\& M : (x * y) - c$$

where; c = number to be factored(d) - 1

$$\text{Let Alli-Smith Expression (A.S.E(K))} = (K + 1)/((F * (K - 1)) + 1)$$

where; F is a positive very large number that approaches infinity.

$$S_1 : \text{Sum}[A.S.E(M), \{x,1,d\}, \{y,1,d\}]$$

S₁/4 is the exact number of possible integer parameter combinations of (x * y) = c + 1

$$\begin{aligned} \text{Sum}[A.S.E(M), \{x,1,d\}, \{y,1,d\}] &= \text{Sum}[(F * d * y + (1 - 2 * F) * \\ &\text{PolyGamma}[0, (1 + F * (-1 - c + y))/(F * y))] + (-1 + 2 * F) * \\ &\text{PolyGamma}[0, (1 + F * (-1 - c + y + d * y))/(F * y))]/(F^2 * y), \{y,1,d\}] \\ &- \text{Expression}(1) \text{ (Thanks to Mathematica \& Stephen Wolfram).} \end{aligned}$$

$$\begin{aligned} \text{Let; Expression}(T) &= (F * d * y + (1 - 2 * F) \text{PolyGamma}[0, (1 + F * \\ &(-1 - c + y))/(F * y)] + (-1 + 2 * F) \text{PolyGamma}[0, (1 + F * (-1 - c + y \\ &+ d * y))/(F * y))]/(F^2 * y) \end{aligned}$$

When we consider the component functions of Expression(T) above we can observe that as variable F approaches infinity, all such functions except for $((1 - 2 * F) * \text{Polygamma}[0, (1 + F * (-1 - c + y))/(F * y))]/(F^2 * y)$ should be ignored as they are limited to

0.

$T_1(F) = (F * d * y)/(F^2 * y)$; as F approaches Infinity, T_1 asymptotically equals 0.

$T_2(F) = ((1 - 2 * F) * \text{Polygamma}[0, (1 + F * (-1 - c + y))/(F * y)])/(F^2 * y)$; as F approaches Infinity, T_2 variates significantly from -Infinity to + Infinity as the value of y changes, since $(1 + F * (-1 - c + y))/(F * y)$ is a negative integer value for various values of y , and $\text{Polygamma}[0, -z]$ is expressed in part by $\text{Cot}[\text{Pi } z]$ function which is $(+/-)\text{Infinity}$ at integer values of z .

$T_3(F) = ((-1 + 2 * F) * \text{PolyGamma}[0, (1 + F * (-1 - c + y + d * y))/(F * y)])/(F^2 * y)$; as F approaches Infinity, T_3 asymptotically equals 0.

**ALLI-SMITH SIGNAL FACTORIZATION
METHOD(A.S.S.F.M)**

To find the factors of the number d from 1 to that number, we convert the number using Expression (1) above into a signal function where all the factor's combination of the number d ; ($x * y$), have their summation value equal to 2, and other non-factor's combination have their summation value asymptotically equalling 0, then we implement a basic binary search [1] to find the factors of the number d .

First Search = $\text{Sum}[\text{Sum}[\text{A.S.E}(M), \{x, 1, N_1\}], \{y, N_1, d\}]$ where;
 $N_1 = \text{Floor}[\text{Sqrt}[d]]$

Second Search = $\text{Sum}[\text{Sum}[\text{A.S.E}(M), \{x, 1, N_1\}], \{y, N_1, d/2\}] \mid \text{Sum}[\text{Sum}[\text{A.S.E}(M), \{x, 1, N_1\}], \{y, d/2, d\}]$, depending on the result of the first search above.

Nth Search = $\text{Sum}[(Nth - 1) \text{ Search}, \{1/2 * (Nth - 1) \text{ Left Interval}\}] \mid \text{Sum}[(Nth - 1) \text{ Search}, \{1/2 * (Nth - 1) \text{ Right Interval}\}]$, depending on the result of the $(Nth - 1)$ search above.

RIEMANN HYPOTHESIS

A number d is considered to be a prime number if the value of it's $S_{1/4}$ expression above is 1.

We consider an expression;

$$R.H.E(d) = \text{Sum}[A.S.E(M), \{x,1,d\}, \{y,1,d\}]/4$$

The distribution of prime numbers [2], from integers; 1,2,3 to an arbitrary integer number R_n can be better observed by considering a series $R.H.E(1), R.H.E(2), R.H.E(3), R.H.E(R_n)$.

$$1, 2, 3, \dots, R_n \leftrightarrow R.H.E(1), R.H.E(2), R.H.E(3), \dots, R.H.E(R_n). - \text{Expression}(2)$$

With further utilization of A.S.E, we can convert the $R.H.E(d)$ series on the right hand side of Expression(2) above into a prime number signal series where the prime numbers from 1,2,3, to an arbitrary integer number R_n all have the value 1 and all non-prime numbers of this series have their value asymptotically equalling 0.

$$P.S.S[1, R_n] = A.S.E(R.H.E(1))/2, A.S.E(R.H.E(2))/2, A.S.E(R.H.E(3))/2, \dots, A.S.E(R.H.E(R_n))/2$$

Where;

P.S.S = Prime Number Signal Series

Using the prime number distribution/theorem formula it is implied that the number of prime numbers from 1 to an arbitrary integer number R_n can be obtained by;

$$R_n / \text{Log}[R_n]$$

We can get a better prime number distribution formula by utilizing the P.S.S signal series above;

$$\text{Sum}[P.S.S[1, R_n]] - \text{Expression}(3)$$

The value of Expression 3 above is the exact number of prime numbers from 1 to any arbitrary integer number R_n .

DIOPHATINE EQUATIONS

Diophatine Equations [3], with arbitrary number of variables can also be resolved easily using A.S.E, to find the missing integer solutions for each variable.

Let B_1 : Diophatine Equation of form $(a^{n_1} + b^{n_2} + \dots + z^n) - c = 1$

& $M_1 : (a^{n_1} + b^{n_2} + \dots + z^n) - c$

where;

c = value of the diophatine equation B_1 above(v) - 1

& n_1, n_2, \dots, n = powers of the variables of the diophatine equation.

$S_2 : \text{Sum}[A.S.E(M_1), \{a, -p, +p\}, \{b, -p, +p\}, \dots, \{z, -p, +p\}]$
 $= \text{Sum}[\text{Sum}[A.S.E(M_1), \{a, -p, +p\}, \{b, -p, +p\}, \dots, \{(z - 1), -p, +p\}], \{z, -p, +p\}] - \text{Expression}(4)$

where; $-p$ to $+p$ is an arbitrary interval.

$S_2/4$ is the exact number of possible integer parameter combinations of $(a^{n_1} + b^{n_2} + \dots + z^n) = c + 1$, that lie within the interval $-p$ to p .

ALLI-SMITH SIGNAL DIOPHATINE

METHOD(A.S.S.D.M)

To find the value of the variables of any diophantine equation, we convert the diophantine equation using Expression(4) into a signal function where all the integer solutions of the diophantine equation have their summation value equal to 2 and all other non-integer solutions of the diophantine equation have their summation value asymptotically equalling 0, then we implement a basic binary search to find the integer solutions of the diophantine equation.

First Search = $\text{Sum}[\text{Sum}[\text{A.S.E}(\text{M}_1), \{a, -p_1, +p_1\}, \{b, -p_2, +p_2\}, \dots, \{(z-1), -p_3, +p_3\}, \{z, -p_n, +p_n\}]$

Where $(+|-)p_2, (+|-)p_3, (+|-)p_n$ are varying intervals.

Second Search = $\text{Sum}[\text{Sum}[\text{A.S.E}(\text{M}_1), \{a, -p_1, +p_1\}, \{b, -p_2, +p_2\}, \dots, \{(z-1), -p_3, +p_3\}, \{z, -p_n, 0\}] \mid \text{Sum}[\text{Sum}[\text{A.S.E}(\text{M}_1), \{a, -p_1, +p_1\}, \{b, -p_2, +p_2\}, \dots, \{(z-1), -p_3, +p_3\}, \{z, 0, +p_n\}]]$, depending on the result of the first search above.

Nth Search = $\text{Sum}[(\text{Nth} - 1) \text{ Search}, \{1/2 * (\text{nth} - 1) \text{ Left Interval}\}] \mid \text{Sum}[(\text{Nth} - 1) \text{ Search}, \{1/2 * (\text{nth} - 1) \text{ Right Interval}\}]$, depending on the result of the $(\text{Nth} - 1)$ search above.

CONCLUSION

The time complexity [4], of the Alli-Smith Signal Factorization Method above is $O(\text{Log}[2, d])$, and the time complexity of the Alli-Smith Signal Diophantine Method above is $O(\text{Log}[2, 2p])$, and both methods can easily be implemented on a classical computer [5], seeing as we only need to resolve summations of functions (hopefully this shouldn't be too hard seeing as the devil "sealest up the sums" lol, Ezekiel 28 vs 12), whilst implementing the binary search for both methods.

ACKNOWLEDGEMENT

I would like to acknowledge and thank Alpha & Omega, Lord God Almighty, for helping me with the knowledge and guidance for this research paper. Without whom we can't even lift a pen to paper, thoughtless of forming and expressing our thoughts via words.

REFERENCES

1). Anthony Lin. (2019). Binary search algorithm.

2). Brian Conrey. (April 2019). Riemann Hypothesis.

3). Bogdan Grechuk. (Aug 2021). Diophantine equations: a systematic approach.

4). Firdous Ahmad Mala. Rouf Ali. (Jan 2022). The Big-O of Mathematics and Computer Science.

5). Qiyu Liu. (March 2023). Comparisons of Conventional Computing and Quantum Computing Approaches.