A Primorial Approach for Generating Primes

Illinois Wesleyan Universtiy
David Lopez | Yash Thacker

Problem | Motivation

Primorials have intrigued mathematicians for centuries because of their historical significance when used by Euler to prove that there are an infinite number of primes. Fascinated by this, we decided to look into their properties, and we discovered a way of Generating prime numbers of a given bit size. This has important applications in many fields.. Prime numbers of a desired bit size , they are essential for making cryptography keys for secure data transmission in cryptosystems like the RSA.

Primorials and Generalized Primorial Primes

A **primorial** is defined as the product of the first n primes :

$$p_n\# \equiv \prod_{k=1}^{n} p_k$$

Generalized primorial primes are defined as prime numbers that are a prime away from a multiple of a primorial.

$$p_n\# \pm (\text{prime or } 1)$$

PPAGP Algorithm

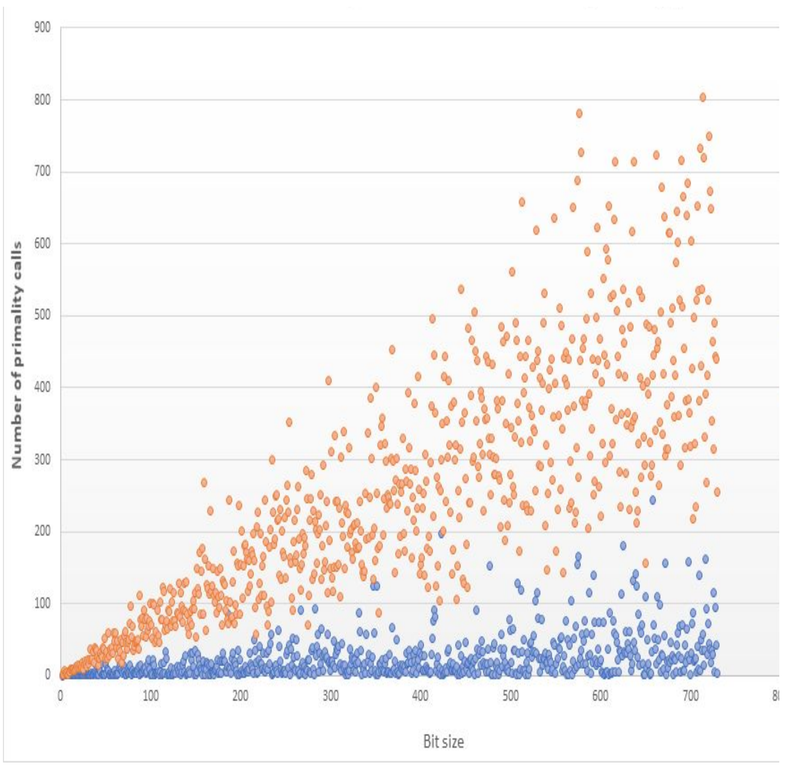
The PPAGP Algorithm generates primes of a given bit size by using Schinzel's conjecture to find **generalized primorial primes**. It loops through

Pseudocode:

1. Generate nth multiple of a primorial given the desired bit size with the form : 2^k and call it "P"
2. **w** $p_n\#$: "P" does not equal prime:
   1. add the (nth +1) prime or 1 if the number of the loop is 1.
   2. if "P" is prime:
   return "P"
   else
   "n" = "n"+1

Computational Time Comparison

Standard method:
PPAGP Algorithm:



Current Solution

To generate prime numbers given a bit size, algorithms generate random numbers within given bounds and test for primality until a number is found. Theory predicts that the number of primality calls is proportional to the bit size.

Pseudocode:

while: "p" does not equal prime:
1. Generate **random** number such that "p" is "n" bits
2. if "p" is prime:
return "p"

Schinzel's Conjecture

Schinzel predicts that:

$$a \cdot p_n\# + c = composite$$

if:

$$c > 1 \text{ and } c < p_{n+1}$$

or

$$c > p_{n+1} \text{ and } c < (p_{n+1})^2 \text{ and } c \text{ is } composite$$

This predicts that we will be able to find a prime near a primorial within a range that is always one or a prime away from the primorial

Conclusions

The PPAG algorithm generates primes ~90% faster than the current current solution for this problem. Currently, it can be applied to key-exchange protocols that do not require randomness, like the Diffie-Hellman key exchange protocol. Moreover, future work could include looking for methods to randomize the algorithm, by looking at modifying the coefficient 2^k.

References:

Stephen P, Richards, A Number For Your Thoughts, 1982, p. 200.
S.W. Golomb Evidence of Fortunes conjecture. Mathematics Magazine, Vol. 54, No. 4 (Sep., 1981), pp. 209-210.