The Multiple Polynomial Quadratic Sieve

Liu Chang, Zhan Dongchen, Zhan Shihan and Zhang Ziyi

I. The Quadratic Sieve

The Multiple Polynomial Quadratic Sieve is mostly constructed on the basis of the original quadratic sieve (QS). Therefore, as a variant of the QS, MPQS could be easily introduced following the QS.

A. Basic of the Quadratic Sieve

The quadratic sieve (QS) is one of the fastest algorithms that find non-trivial factors of a large composite numbers. Actually, the QS is the fastest known factoring algorithm for large numbers with up to 110 digits till today.

Given the number n to be factored, the QS attempts to find two integers x and y such that

$$x \not\equiv \pm y \bmod n, \quad x^2 \equiv y^2 \bmod n.$$

It is guaranteed that

$$(x-y)(x+y) \equiv 0 \bmod n.$$

Hence, it follows that $\gcd(x-y,n)$ is probably a non-trivial factor of n. Further study shows that the probability of $\gcd(x-y,n)$ being non-trivial is at least $1/2$, which is fairly promising and efficient. Note that $\gcd(x-y,n)$ can be efficiently calculated by applying the Extended Euclidean Algorithm. Thus, the major problem now is how to find a sufficient number of such integers x and y, so that non-trivial factors of the number n could be found with an almost surely probability.

First, we define a quadratic function

$$Q(x) = (x + \lfloor\sqrt{n}\rfloor)^2 - n = \tilde{x}^2 - n.$$

Note that for all x, we have

$$Q(x) \equiv \tilde{x}^2 \bmod n.$$

Thus, by selecting a set of x, such that the product of $Q(x)$, $Q(x_1)Q(x_2)\cdots Q(x_r)$, is also a square, denoted by y^2, i.e.,

$$Q(x_1)Q(x_2)\cdots Q(x_r) \equiv y^2 \bmod n.$$

Since we have each $Q(x)$ as a product of $\tilde{x}$, then we obtain

$$Q(x_1)Q(x_2)\cdots Q(x_r) \equiv (x_1x_2\cdots x_r)^2 \bmod n.$$

Now, we construct the quadratic equation in the form of $y^2 \equiv x^2 \bmod n$, and the rest of the problem is solvable by applying the Extended Euclidean Algorithm. Hence, the problem is how to select ("sieve") such set of x_i.

B. Factor Base and Sieving Interval

Given the mathematical foundation of the quadratic sieve, an efficient way to select the set of x_i is still needed.

One of the most essential requirements for $Q(x_i)$ is that the product of $Q(x_i)$ must be a square, and thus, the powers of all the prime factors of the product $Q(x_1)Q(x_2)\cdots Q(x_r)$ need to be even.

Given a set of x_i, to examine if our corresponding product $Q(x_1)Q(x_2)\cdots Q(x_r)$ satisfies the aforementioned requirement, we need to factor each of the $Q(x_i)$. To avoid suffering from factoring large $Q(x_i)$, we want all $Q(x_i)$s to be small, and can be factored over a fixed set of small prime numbers, including -1. The fixed set of small prime numbers that all factor the product $Q(x_1)Q(x_2)\cdots Q(x_r)$ is called *factor base*.

Since $Q(x_i)$s are defined to be a quadratic function of x_i, to make $Q(x_i)$ small, we need x_i to be small enough. Hence, we need each x_i to be selected close to 0 by setting a bound M. Only values under the bound, i.e., $[-M, M]$, are considered as the candidates of x_i. The range of selecting x_is are called *sieving interval*.

Consider x, a value in the sieving interval, and

$$Q(x) = (x - \sqrt{n})^2 - n.$$

If some prime p is a prime factor of $Q(x)$, i.e., $p|Q(x)$, then

$$(x-\sqrt{n})^2 - n \equiv 0 \bmod p,$$

$$(x-\sqrt{n})^r \equiv n \bmod n.$$

It immediately follows that n is a square (quadratic residue) $\bmod n$, writing by the Legendre symbol as follows

$$\left(\frac{n}{p}\right) = 1.$$

Therefore, all the primes p in the factor base satisfy the Legendre symbol above.

The other requirement for the primes is that all the primes are expected to be small, and thus again, are under some certain bound B. The size of the bound B is dependent on the size of the number n to be factored.

C. Sieving

By introducing an appropriate factor base and a proper sieving interval, what we need to do is essentially taking numbers x as trials from the sieving interval, calculating the corresponding $Q(x)$, and examine whether $Q(x)$ factors completely over the introduced factor base. In this step, for simplicity, we define the concept of smoothness in the following definition.

Definition I.1 (Smoothness)**.** *A number $Q(x)$ is said to be smooth, or have smoothness, if it factors completely over the adopted factor base.*

Notice that we want to keep trace of the powers of all the prime factors of the product $Q(x_1)Q(x_2)\cdots Q(x_r)$, and thus we must keep trace of the powers of all the prime factors of each $Q(x_i)$. Therefore, if $Q(x)$ of the current trial x does not have smoothness, we leave the trial x out of consideration, and go on to the next integer in our adopted sieving interval. If $Q(x)$ is smooth, then the current trial x is kept as a candidates of elements in the set of x_i.

Now, we could again justify the reason why we need to introduce a relatively small factor base where all prime factors are bounded by B. If the factor base is large, then the probability of having a smooth $Q(x)$ is unacceptably small, and even worse, the value of x needs to be increased accordingly, which would lead to computational inefficient or even infeasibility.

However, even with a fairly small factor base, it is still inefficient to consider numbers in the sieving interval one at a time and examine the smoothness over the entire factor base for each trial x.

Hence, it is necessary that we select and examine x in the sieving interval in parallel, with each processor working on a different and mutually exclusive subinterval of the sieving interval.

D. Sieving in Parallel

Now we consider how to realize sieving in parallel.

If p is a prime factor of $Q(x)$, then by noticing

$$Q(x+p) = \left((x-\sqrt{n})+p\right)^2,$$

it follows

$$p | Q(x+p).$$

Conversely, if

$$x \equiv y \bmod p,$$

then we have

$$Q(x) \equiv Q(y) \bmod p.$$

For each prime p in the adopted factor base, we solve

$$Q(x) = s^2 \equiv 0 \bmod p, \quad x \in \mathbb{Z}_p$$

by applying the Shanks-Tonelli Algorithm. Two solutions will be obtained given by the Shanks-Tonelli Algorithm, denoted by s_{1p} and $s_{2p} = p - s_{1p}$. Now, by finding s_{1p} and s_{2p} that could be factored by p, it could immediately be obtained that all x_i that are congruent modulo p will be factor by p, i.e.,

$$x_i = s_{1p}, s_{2p} + pk, \quad k \in \mathbb{Z}.$$

Select a subinterval of the sieving interval, and record $Q(x_i)$ in an array for each x_i in the subinterval. For each prime factor p in the factor base, we start at s_{1p} and s_{2p}. Find the highest power of the prime factor p by keeping dividing out from $Q(x_i)$. Given the highest power of the prime factor p for $Q(x_i)$, calculate its value modulo 2 in a vector.

Now, we have an array of $Q(x_i)$ for all each of the elements x_i in a subinterval of the sieving interval. Furthermore, for each entry $Q(x_i)$ in the array, we keep a vector, with the length equals to the number of prime factors in the factor base, where each entry corresponds to the highest power modulo 2 of a unique prime factor that can be divided out of $Q(x_i)$.

After being divided by all the primes in the factor base, the number recorded in the array at the position of $Q(x_i)$ would now be 1 if and only if $Q(x_i)$ has smoothness.

Therefore, we could keep all the $Q(x_i)$s which are now 1 in the array and throw out the rest. The vector, which records the highest power modulo 2, could be written in a matrix A, where each column corresponds to a distinct $Q(x_i)$. Repeat the sieving procedure, until sufficient columns are present in the matrix A.

E. The Matrix A

As described in the previous section, only if $Q(x_i)$ is smooth, could the corresponding vector be put into the matrix A. Therefore, to obtain the product of the $Q(x_i)$s as a perfect square, we need to find such a set of $Q(x_i)$s that the sum modulo 2 of each row of the selected $Q(x_i)$s are 0. Essentially, we are selecting in this way to ensure that the power of each prime factor of the final product $Q(x_1)Q(x_2)\cdots Q(x_r)$ are even, and thus, the product is a square modulo n.

Given the matrix A, there are various ways in terms of selecting a set of $Q(x_i)$, such that the aforementioned requirements are satisfied. The objective written in a mathematical form is that, given a set of $Q(x_1), Q(x_2), \cdots, Q(x_k)$, a solution in the form

$$Q(x_1)e_1 + Q(x_2)e_2 + \cdots + Q(x_k)e_k,$$

where e_is are boolean variables taking values in $\{0,1\}$, and $\mathbf{a}_i$ being the ith column of the matrix A corresponding to $Q(x_i)$. We want $\mathbf{a}_1 e_1 + \mathbf{a}_2 e_2 + \cdots + \mathbf{a}_k e_k \equiv \mathbf{0} \bmod 2$, or equivalently

$$\mathbf{e}A = \mathbf{0} \bmod 2,$$

where

$$\mathbf{e} = (e_1, e_2, \cdots, e_k).$$

The above equation could be efficiently solved by applying the *Gaussian elimination*, via which the spanning set of the solution space could be found.

II. Multiple Polynomial Quadratic Sieve (MPQS)

Very much similar to the original quadratic sieve, the only difference of the Multiple Polynomial Quadratic Sieve from the original QS is that, instead of using a single quadratic function $Q(x)$ in the algorithm, MPQS introduces multiple polynomials.

All the polynomials in the MPQS are in the form

$$Q(x) = ax^2 + 2bx + c.$$

The parameters a, b, c are chosen with the following criterion.

- $0 \leq b < a$, and $b^2 \equiv n \bmod a$.

- for every $q|a$, $\left(\frac{n}{q}\right) = 1$.
- $b^2 - 4ac = n$.

Now, we could notice that $Q(x)$ could be written in the following form:

$$(ax+b)^2 \equiv aQ(x) \bmod n.$$

Since a is chosen to be a square, then $Q(x)$ is guaranteed to be a square as well. Suppose the sieving interval is $[-M, M]$. Then, we usually construct a in order to have the same absolute values of maximum and minimum on this interval. The minimum on this interval is when $x = -\frac{b}{a}$, and

$$Q(-\frac{b}{a}) = a \cdot (-\frac{b}{a})^2 - 2b \cdot \frac{b}{a} + c = -\frac{n}{a}$$

Also, the maximum value is about$\frac{a^2M^2-n}{a}$. After assigning a equation to these two terms, we have

$$a = \frac{\sqrt{2n}}{M}$$

Hence, in the aforementioned manner, we could find a sufficient number of such x_is and corresponding $Q(x_i)$s that hopefully at least one non-trivial factor of n is found via

$$y^2 \equiv x^2 \bmod n, \quad x \not\equiv \pm y \bmod n.$$

The advantage of adopting such a variant MPQS of the original QS is that by using multiple polynomials, the size of the factor base and the sieving interval could be significantly reduced, which helps decrease the running time of the sieving algorithm. A good estimate for the optimal size of the factor base is about a tenth as many as the size of the factor base needed for the original QS. It indicates that the size of the size of the sieving interval used for MPQS is about $1/1000$ of the size of sieving interval needed for the original QS.

III. Running Time Analysis

For a large integer, it is likely that we only need a small base containing a few primes to factor this integer. In this way, we need a great deal of time to factor when integers become large. Therefore, we need to obtain the time complexity of each matrix. The complexity of optimal for the factor base is

$$B = (e^{\ln{(n)}\ln{(\ln{(n)})}})^{\sqrt{2}/4}$$

The complexity of the sieving interval M is the cubic of B.

$$M = (e^{\ln{(n)}\ln{(\ln{(n)})}})^{3\sqrt{2}/4}$$

Further research has been done later and the time complexity is improved by Gaussian Elimination. For now, the fastest public algorithm of QS is the Number Field Sieve.

$$O(e^{1.9233(\ln n)^{1/3}(\ln{(\ln{(n)})})^{2/3}})$$

References

- Eric Landquist, The Quadratic Sieve Factoring Algorithm, http://www.cs.virginia.edu/crab/QFS_Simple.pdf, 2001.