PROBABILISTIC APPROACH IN FACTORIZATION BY ELLIPTIC CURVE METHOD

I. Dermenzhy, G. Vostrov
Odessa national polytechnic university
Odessa, UKRAINE
e-mail: ivandermenji97@gmail.com, vostrov@gmail.com

Abstract

The aim of this work is the analysis of the elliptic curve factorization method by using probabilistic number theory apparatus. Particularly, the relationship between the amount generated curves and the required boundary problem research.

Keywords: elliptic curve, factorization, probabilistic number theory

1 Introduction

A new probabilistic number theory approach is defined by Erdos-Kac theorem [2] and E. Kowalski point of view about it [3]. Erdos-Kac theorem connects the distribution of the different large numbers prime divisors with the probability theory limit laws formulas.

Theorem 1. *(The Erdos-Kac theorem). For any $n \geq 1$, if $\omega(n)$ is the amount of given number different prime divisors, then for any real numbers $a < b$:* $\lim\limits_{N\to+\infty} \frac{1}{N}|\{1 \leq n \leq N | a \leq \frac{\omega(n)-\log\log N}{\sqrt{\log\log N}} \leq b\}| = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-x^2/2} dx$ *is satisfied.*

That is, the limit distribution of $\frac{\omega(n)-\log\log N}{\sqrt{\log\log N}}$, corresponds to the standard normal distribution [2]. Hence the conclusion, that the amount of natural numbers n with a small number of divisors is low, and the amount of natural numbers with a great number of divisor is too.

This fact leads us to change the approach of factorization problem consideration. It is possible that, based on the theorem 1, it is necessary to investigate and deepen factorization methods based on probabilistic number theory. This approach, in turn, seems quite natural also due to the fact that the best factorization methods, which belong to the sub-exponential class, have probabilistic character.

2 Stochastic elliptic curves method

Among this class of factorization methods, we should single out method based on the elliptic curves theory. This method provides the greatest interest due to a number of features: the dependence of the method computational complexity primarily on the factorized number smallest prime divisor, many ways of optimization, possibility of parallelization.

However ECM has a list of problems, such as the optimal number of curves after using which it would be necessary to increase the method boundary problem. Correct choice of this number can lead to a significant reduction in the time costs of the method software implementation, as shown further.

It is important to estimate the dependence of the method computational complexity as a function of the chosen boundary. First, we need to know the number of primes on a given interval. The prime number theorem does not give exact estimation. Estimating of prime numbers degrees – α_i values, is not easy task either. All this is complicated by the probabilistic nature of the method.

Proceeding from the foregoing, a theoretical assessment of this relationship, even if it is possible, will be heuristic in view of the huge number of uncertainties. Thus, it makes sense to use an empirical estimate.

Software that modeling the factorization process based on the theory of elliptic curves was implemented. This software is based on the idea of the algorithm proposed by Pomerance [1]. The input of the method software implementation is composite numbers consisting of the two prime numbers product with a size of $\sim 10^5$. For the representativeness of the obtained results, for each case, 10 such composite numbers were used as input, each of which was factorized 30 times. The cases with the initial boundary: 100, 30, 6, 2, 1 were considered, these boundaries where chosen empirically respectively to the size of factorized numbers. Obviously we can't get 1-smooth number, however when choosing boundary equal 1, we still got chance to factorize this number by getting divisor as G.C.D. of curve's discriminant and factorized number. Such boundaries were chosen in the research process due to the gradual acceleration of the software with a decrease of the initial boundary. Numbers were taken from the interval, in increments of 500, as the number of curves used, after which the boundary is increased. This is due to the fact that at step 500, obtained results fairly accurately describe the overall efficiency of the method, while this interval length is quite enough for a clear fixation of trend.

The dependence of the time spent on the software implementation work, the final boundary and the number of curves used depending on the number of curves after which the boundary is increased was investigated. Results are reflected in Fig. 1, Fig. 2 and Fig. 3 respectively.

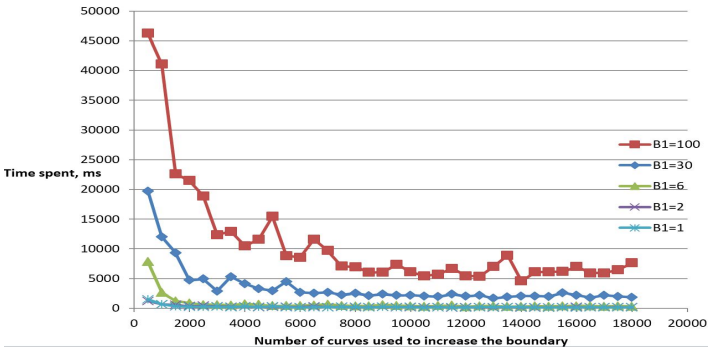

Figure 1: The graph of the time consumed dependence on the number of curves, after which the boundary is increased

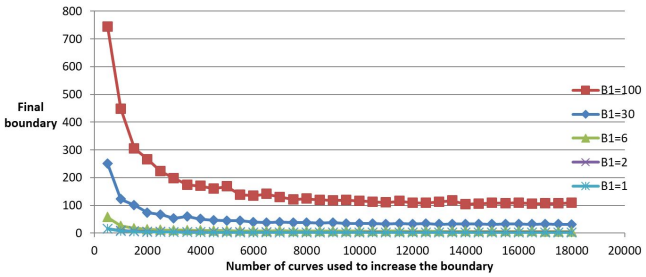

Figure 2: The graph of the final boundary dependence on the number of curves, after which the boundary is increased

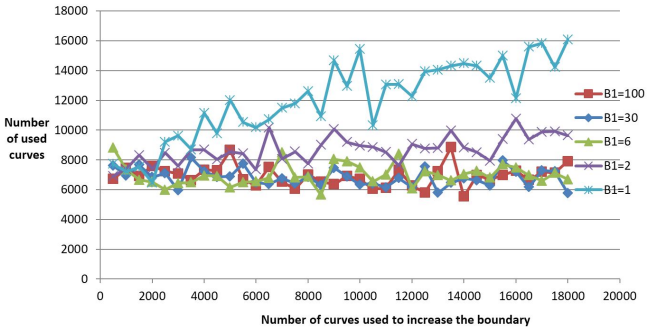

Figure 3: The graph of the curves number used for factorization dependence on the number of curves, after which the boundary is increased

As can be seen from Figures 1 and 2, for all the cases considered, an increase in the number of curves required to change boundary led to a decrease in time costs, as well as to a decrease in the final boundary value. The final boundary with the increase in the number of curves required for changing the boundary converged to its original value. Thus, for numbers whose divisors consist of 5 decimal characters, it is almost always possible to get a curve of b_1-smooth order, or the fact that in this case $1 < g < n$, where $g = G.C.D.\,(4a^3 + 27b^2)$ is satisfied. In such case, the discriminant of the generated curve has a common divisor with a composite number n, different from 1.

It's unlikely to obtain a curve of 1-, 2- or 6-smooth order, since it is a very strict condition. Therefore, most likely for these cases, the second condition was satisfied. Based on this, the percentage of the cases number in which the second condition was satisfied was estimated (Figure 4).

Figure 4 shows, that for boundary $b_1 = 100$ percentage of such cases fluctuates in a range from 8% to 20%. In case, when $b_1 = 30$ – from 19% to 34%, when $b_1 = 6$ – from 28% to 38%. In cases when $b_1 = 2$ and $b_1 = 1$ the probability of getting b_1-smooth order curve is the lowest, as evidenced by the highest percentage of cases when the divisor was detected using the curve discriminant. The percentage of such situations ranges from 31% to 52% and 32% to 83% respectively. In addition, for the last two boundaries the frequency of cases when the divisor was found by using a discriminant increases with the number of curves, after which the boundary increases. This is due

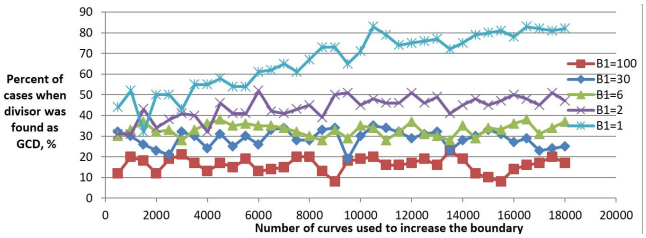

Figure 4: The graph of the finding the divisor by a discriminant of curve dependence on the number of curves, after which the boundary is increased

to the fact that, if these boundaries remain unchanged, the probability that we will get a curve of B_1-smooth order, is much lower, so we will rather find a curve with the suitable discriminant. This requires the use of a larger curves amount, as shown in Figure 3. Thus, for first three cases, the total number of curves used ranges from 6000 to 9000, and their average value is close for all three cases. In the last two cases, the average number of curves used is much greater.

So, for numbers whose smallest divisor is around 10^5 and less, it is much more advantageous to increase the number curves used than the boundary to reduce the time required for factorization process, even in cases of a minimal boundary. This is possible due to the probability that the discriminant of the curve has a common divisor with the specified composite number. This is indicated by the results obtained when $b_1 = 1$, was taken as initial boundary. In this case the percentage of these cases was about 80%, while the time costs were the lowest. This is due to a significant increase in the algorithm complexity with the growth of the boundary b_1, much greater than with increasing of the used curves number for given composite numbers.

However, we cannot exclude the cases when the curves whose order was b_1- smooth were received. These cases occurred for $b_1 = 2$, and for $b_1 = 1$, providing a good executing time of program realization. Also despite the result we can't really implement them very far. The reason is that we can't establish the fact that composite number contains divisors less than 10^5.

The represented results are only the first steps in solving of used curves limit after which boundary should be increased problem.

References

[1] Crandall R.E., Pomerance C. B. (2006). *Prime numbers: A Computational Perspective*. Springer-Verlag, New-York.

[2] Erdos P., Kac M. (1940). The Gaussian Law of Errors in the Theory of Additive Number Theoretic Functions. *American J. Math.* Vol. **62**, pp. 738-742.

[3] Kowalski E., (2018). Arithmetic Randonnee, An introduction to probabilistic number theory. Available at: https://people.math.ethz.ch/ kowalski/probabilistic-number-theory.pdf