МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52 DOI 10.17223/20710410/61/7

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ ФАКТОРИЗАЦИИ ЦЕЛЫХ ЧИСЕЛ[1]

А. Н. Рыбалов

Институт математики им. С. Л. Соболева СО РАН, г. Омск, Россия

E-mail: alexander.rybalov@gmail.com

Изучается генерическая сложность проблемы факторизации целых чисел. Данная проблема, восходящая ещё к Гауссу, имеет важное значение для современной криптографии. Например, на предположении о её трудноразрешимости основывается криптостойкость системы шифрования с открытым ключом RSA. В работе доказывается, что при условиях трудноразрешимости этой проблемы в худшем случае и P = BPP для её решения не существует полиномиального сильно генерического алгоритма. Сильно генерический алгоритм решает проблему не на всём множестве входов, а на подмножестве, последовательность относительных плотностей которого при увеличении размера экспоненциально быстро сходится к единице. Для доказательства используется метод генерической амплификации, который позволяет строить генерически трудные проблемы из проблем, трудных в худшем случае. Основным ингредиентом этого метода является объединение эквивалентных входов в достаточно большие множества. Эквивалентность входов означает, что рассматриваемая проблема на них решается одинаково.

Ключевые слова: *генерическая сложность, факторизация целых чисел.*

ON GENERIC COMPLEXITY OF THE INTEGER FACTORIZATION PROBLEM

A. N. Rybalov

Sobolev Institute of Mathematics, Omsk, Russia

In the paper, we study the generic complexity of the integer factorization problem. This problem, which goes back to Gauss, has important applications in modern cryptography. For example, the cryptographic strength of the famous public key encryption system RSA is based on the assumption of its hardness. We prove that under the condition of worst-case hardness and P = BPP for the problem of integer factorization there is no polynomial strongly generic algorithm. A strongly generic algorithm solves a problem not on the entire set of inputs, but on a subset whose frequency sequence converges exponentially to 1 with increasing size. To prove this theorem, we use the method of generic amplification, which allows to construct generically hard problems from the problems hard in the classical sense. The main component of this method is

[1]Работа выполнена в рамках госзадания ИМ СО РАН, проект FWNF-2022-0003.

the cloning technique, which combines problem inputs together into sufficiently large sets of equivalent inputs. Equivalence is understood in the sense that the problem is solved similarly for them.

Keywords: *generic complexity, integer factorization.*

Введение

Проблема факторизации (разложения на множители) целых чисел является классической алгоритмической проблемой теории чисел, восходящей ещё к Гауссу. Для неё до сих пор не найдены эффективные (полиномиальные) алгоритмы [1]. На предположении о трудноразрешимости проблемы факторизации основана знаменитая система шифрования с открытым ключом RSA [2].

В современной криптографии интересны такие проблемы, которые, являясь (гипотетически) трудными в классическом смысле, остаются трудными и в генерическом смысле, т. е. для почти всех входов. Это объясняется тем, что при случайной генерации ключей в криптографическом алгоритме происходит генерация входа некоторой трудной алгоритмической проблемы, лежащей в основе алгоритма. Если проблема будет легкоразрешимой почти всегда, то для почти всех таких входов ее можно будет быстро решить и ключи почти всегда будут нестойкими. Поэтому проблема должна быть трудной для почти всех входов. Например, таким поведением обладают классические алгоритмические проблемы криптографии: проблема распознавания квадратичных вычетов [3], проблема дискретного логарифма [4], проблема извлечения корня в группах вычетов [5] (проблема обращения функции RSA).

Генерический подход [6] — это один из подходов к изучению алгоритмических проблем для «почти всех» входов. В рамках этого подхода алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. Такие входы образуют так называемое генерическое множество. Понятие «почти все» формализуется введением естественной меры на множестве входных данных. С точки зрения практики алгоритмы, решающие быстро проблему на генерическом множестве, так же хороши, как и быстрые алгоритмы для всех входов.

В данной работе изучается генерическая сложность проблемы факторизации целых чисел. Доказывается, что при условии трудноразрешимости этой проблемы в худшем случае и P = BPP для неё не существует полиномиального сильно генерического алгоритма. Сильно генерический алгоритм решает проблему не на всём множестве входов, а на подмножестве, последовательность относительных плотностей которого при увеличении размера экспоненциально быстро сходится к единице. Класс BPP состоит из проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. Считается, что класс BPP совпадает с классом P, то есть любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, построив полиномиальный алгоритм, не использующий генератор случайных чисел и решающий ту же самую проблему. Хотя равенство P = BPP до сих пор не доказано, имеются веские основания в пользу него [7].

1. Генерические алгоритмы

Пусть I — некоторое множество входов. Для подмножества $S \subseteq I$ определим *последовательность относительных плотностей*

$$\rho_n(S) = \frac{|S_n|}{|I_n|},\ n = 1, 2, 3, \ldots,$$

где I_n — множество входов размера n, а $S_n = S \cap I_n$. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. В данной работе множеством входов для алгоритмов является множество натуральных чисел, записанных в двоичной форме. Под размером натурального числа понимаем длину его двоичной записи.

Асимптотической плотностью множества S назовем верхний предел

$$\rho(S) = \overline{\lim_{n\to\infty}} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$, и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо.

Следуя [6], назовём множество S *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к нулю, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что для любого n

$$\rho_n(S) < C\sigma^n.$$

Теперь S называется *сильно генерическим*, если его дополнение $I \setminus S$ сильно пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется (*сильно*) *генерическим*, если:

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) \neq ?\}$ является (сильно) генерическим.

Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если $(\mathcal{A}(x) = y \in J) \Rightarrow$ $\Rightarrow (f(x) = y)$ для всех $x \in I$. Ситуация $\mathcal{A}(x) = ?$ означает, что $\mathcal{A}$ не может вычислить функцию f на аргументе x. Но условие 2 гарантирует, что $\mathcal{A}$ корректно вычисляет f на почти всех входах (входах из генерического множества). Множество $S \subseteq I$ называется (*сильно*) *генерически разрешимым за полиномиальное время*, если существует (сильно) генерический полиномиальный алгоритм, вычисляющий его характеристическую функцию.

2. Вероятностные алгоритмы

Напомним некоторые понятия классической теории сложности вычислений. *Время работы* $t_M(x)$ машины Тьюринга M на входе $x \in I$ — это число шагов машины от начала работы до остановки. Если M на x не останавливается, полагаем $t_M(x) = \infty$. Машина Тьюринга M *полиномиальна*, если существует полином $p(n)$, такой, что для любого $x \in I$ имеет место $t_M(x) < p(|x|)$. Класс P состоит из подмножеств I, распознаваемых полиномиальными машинами Тьюринга.

Вероятностная машина Тьюринга — это машина Тьюринга, в программе которой допускаются пары правил вида

$$(q_i, a) \to (q_j, b, S_1),$$
$$(q_i, a) \to (q_k, c, S_2).$$

В процессе работы такой машины с вероятностью $1/2$ выбирается первое правило и с вероятностью $1/2$ — второе. Обозначим через $\mathsf{P}[M(x) = y]$ вероятность того, что машина M на входе x выдаёт ответ y. Время работы $t_M(x, \tau)$ вероятностной машины Тьюринга на входе x зависит от вычислительного пути (последовательности выполненных команд) τ. Проблема $S \subseteq I$ принадлежит классу BPP, если существует вероятностная машина Тьюринга M и полином $p(n)$, такие, что:

1) для любого x и для любого вычислительного пути τ машины M на x имеет место $t_M(x,\tau) < p(|x|)$;
2) если $x \in S$, то $\mathsf{P}[M(x) = 1] > 2/3$;
3) если $x \notin S$, то $\mathsf{P}[M(x) = 0] > 2/3$.

Вероятностные машины Тьюринга формализуют понятие алгоритма, использующего генератор случайных чисел. Класс BPP — это класс проблем, эффективно решаемых такими вероятностными алгоритмами. Большинство специалистов по теоретической информатике сейчас считает, что имеет место равенство P = BPP. Это означает, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, т. е. построить полиномиальный детерминированный алгоритм, решающий ту же задачу. Хотя равенство пока не доказано, имеются серьёзные результаты в его пользу [7].

3. Проблема факторизации целых чисел

Проблема факторизации целых чисел состоит в следующем. Дано натуральное число N, записанное в двоичной системе. Необходимо найти его разложение в произведение степеней простых чисел: $N = p_1^{k_1} \dots p_m^{k_m}$.

Лемма 1. Существует полиномиальный алгоритм, который для любых натуральных чисел N и M по разложению на простые множители их произведения $NM = p_1^{k_1} \dots p_m^{k_m}$ находит разложение на простые множители отдельно для чисел N и M.

Доказательство. Искомый полиномиальный алгоритм работает следующим образом. Для каждого простого числа p_i, входящего в степени k_i в разложение числа NM, пытаемся разделить N сначала на p_i. Если N делится на p_i без остатка, то N/p_i делим на p_i. И так далее до тех пор, пока не получим ненулевой остаток от деления. Тем самым находим максимальную степень s_i простого p_i, входящую в разложение числа N. Проделав это для всех p_i, найдём искомое разложение $N = p_1^{s_i} \dots p_m^{s_m}$. Тогда $M = p_1^{k_i - s_i} \dots p_m^{k_m - s_m}$.

Полиномиальность алгоритма следует из того, что операция деления с остатком проводится за полиномиальное время и количество простых множителей в разложении числа NM ограничено величиной $\log(NM)$, то есть размером входа. ■

4. Основной результат

Теорема 1. Если существует сильно генерический полиномиальный алгоритм, решающий проблему факторизации целых чисел, то существует вероятностный полиномиальный алгоритм, решающий эту проблему на всём множестве входов.

Доказательство. Допустим, что существует сильно генерический полиномиальный алгоритм $\mathcal{A}$, решающий проблему факторизации. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всём множестве входов. На натуральном числе N размера n ($2^n \leqslant N < 2^{n+1}$) алгоритм $\mathcal{B}$ работает следующим образом:

1) генерирует случайно и равновероятно натуральное число M размера $n^2 - n$;
2) запускает алгоритм $\mathcal{A}$ на числе NM;
3) если $\mathcal{A}(NM) \neq ?$, то есть алгоритм выдаёт разложение $NM = p_1^{k_1} \dots p_m^{k_m}$, то по лемме 1 находим за полиномиальное время разложение на простые множители для числа N;
4) если $\mathcal{A}(NM) = ?$, то выдаёт ответ 2.

Заметим, что полиномиальный вероятностный алгоритм $\mathcal{B}$ выдаёт правильный ответ на шаге 3, а на шаге 4 может выдать неправильный ответ. Нужно доказать, что вероятность того, что ответ выдаётся на шаге 4, меньше $1/3$.

Оценим вероятность выдачи ответа на шаге 4. Число M имеет размер n^2-n, значит, размер числа NM равен $n^2-n+n=n^2$. Вероятность того, что для NM имеет место $\mathcal{A}(NM)=?$, не больше

$$\frac{|\{K\in\mathbb{N}:\mathcal{A}(K)=?\}_{n^2}|}{|\{NM:M\in\mathbb{N}_{n^2-n}\}_{n^2}|}=\frac{|\{K\in\mathbb{N}:\mathcal{A}(K)=?\}_{n^2}|}{|\mathbb{N}_{n^2}|}\,\frac{|\mathbb{N}_{n^2}|}{|\{NM:M\in\mathbb{N}_{n^2-n}\}_{n^2}|}=$$
$$=\frac{|\{K\in\mathbb{N}:\mathcal{A}(K)=?\}_{n^2}|}{|\mathbb{N}_{n^2}|}\,\frac{2^{n^2}}{2^{n^2-n}}=2^n\frac{|\{K\in\mathbb{N}:\mathcal{A}(K)=?\}_{n^2}|}{|\mathbb{N}_{n^2}|}.$$

Так как множество $\{K\in\mathbb{N}:\mathcal{A}(K)=?\}$ сильно пренебрежимое, то существует константа $\alpha>0$, такая, что

$$\frac{|\{K\in\mathbb{N}:\mathcal{A}(K)=?\}_{n^2}|}{|\mathbb{N}_{n^2}|}<\frac{1}{2^{\alpha n^2}}$$

для любого n. Поэтому искомая вероятность ответа на шаге 4 не больше

$$\frac{2^n}{2^{\alpha n^2}}=\frac{1}{2^{\alpha n^2-n}}<\frac{1}{3}$$

при больших n. ■

Теорема 2. Если для проблемы факторизации не существует полиномиального алгоритма и P = BPP, то для неё не существует сильно генерического полиномиального алгоритма.

Доказательство. Пусть существует сильно генерический алгоритм, решающий проблему факторизации. Тогда, по теореме 1, существует вероятностный полиномиальный алгоритм, решающий её на всём множестве входов. Этот же алгоритм решает следующую проблему распознавания A, которая полиномиально эквивалентна проблеме факторизации: даны натуральные числа N и K, нужно определить, существует ли нееединичный множитель N меньше K. Таким образом, проблема A лежит в классе BPP. Так как P = BPP, то она лежит и в классе P, а значит, для проблемы факторизации существует полиномиальный алгоритм. Противоречие. ■

ЛИТЕРАТУРА

1. *Adleman L. M. and McCurley K. S.* Open problems in number theoretic complexity, II // LNCS. 1994. V. 877. P. 291–322.
2. *Rivest R., Shamir A., and Adleman L.* A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM. 1978. V. 21. Iss. 2. P. 120–126.
3. *Рыбалов А. Н.* О генерической сложности проблемы распознавания квадратичных вычетов // Прикладная дискретная математика. 2015. № 2 (28). С. 54–58.
4. *Рыбалов А. Н.* О генерической сложности проблемы дискретного логарифма // Прикладная дискретная математика. 2016. № 3 (33). С. 93–97.
5. *Рыбалов А. Н.* О генерической сложности проблемы извлечения корня в группах вычетов // Прикладная дискретная математика. 2017. № 38. С. 95–100.
6. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
7. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC. El Paso, ACM, 1997. P. 220–229.

REFERENCES

1. *Adleman L. M. and McCurley K. S.* Open problems in number theoretic complexity, II. LNCS, 1994, vol. 877, pp. 291–322.
2. *Rivest R., Shamir A., and Adleman L.* A method for obtaining digital signatures and public-key cryptosystems // Commun. ACM, 1978, vol. 21, iss. 2, pp. 120–126.
3. *Rybalov A. N.* O generitcheskoy slozhnosti problemy raspoznavaniya kvadratichnykh vychetov [On generic complexity of the quadratic residuosity problem]. Prikladnaya Diskretnaya Matematika, 2015, no. 2 (28), pp. 54–58. (in Russian)
4. *Rybalov A. N.* O generitcheskoy slozhnosti problemy diskretnogo logarifma [On generic complexity of the discrete logarithm problem]. Prikladnaya Diskretnaya Matematika, 2016, no. 3 (33), pp. 93–97. (in Russian)
5. *Rybalov A. N.* O generitcheskoy slozhnosti problemy izvlecheniya kornya v gruppakh vychetov [On generic complexity of the problem of finding roots in groups of residues]. Prikladnaya Diskretnaya Matematika, 2017, no. 38, pp. 95–100. (in Russian)
6. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 2003, vol. 264, no. 2, pp. 665–694.
7. *Impagliazzo R. and Wigderson A.* P = BPP unless E has subexponential circuits: Derandomizing the XOR Lemma. Proc. 29th STOC, El Paso, ACM, 1997, pp. 220–229.