МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52 DOI 10.17223/20710410/58/10

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ РАЗБИЕНИЯ ГРАФА НА ТРЕУГОЛЬНИКИ[1]

А. Н. Рыбалов

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

E-mail: alexander.rybalov@gmail.com

NP-полнота проблемы разбиения графа на треугольники доказана Шейфером в 1974 г. и содержится в классической монографии М. Гэри и Д. Джонсона. В данной работе изучается генерическая сложность этой проблемы. Доказывается, что при условии $P \neq NP$ и $P = BPP$ для её решения не существует полиномиального сильно генерического алгоритма.

Ключевые слова: *генерическая сложность, разбиение графа на треугольники.*

THE GENERIC COMPLEXITY OF THE GRAPH TRIANGULATION PROBLEM

A. N. Rybalov

Sobolev Institute of Mathematics, Omsk, Russia

Generic-case approach to algorithmic problems was suggested by A. Miasnikov, V. Kapovich, P. Schupp, and V. Shpilrain in 2003. This approach studies behavior of an algorithm on typical (almost all) inputs and ignores the rest of inputs. In this paper, we study the generic complexity of the graph triangulation problem. This problem is as follows. Given a finite simple graph with $3n$ vertices, determine whether the vertices of the graph can be divided into n three-element sets, each of which contains vertices connected by edges of the original graph (that is, they are triangles). NP-completeness of this problem was proved by Shaffer in 1974 and is mentioned in the classic monograph by M. Garey and D. Johnson. We prove that under the conditions $P \neq NP$ and $P = BPP$ there is no polynomial strongly generic algorithm for this problem. A strongly generic algorithm solves a problem not on the whole set of inputs, but on a subset whose frequency sequence converges exponentially to 1 with increasing size. To prove the theorem, we use the method of generic amplification, which allows one to construct generically hard problems from the problems that are hard in the classical sense. The main component of this method is the cloning technique, which combines the inputs of a problem together into sufficiently large sets of equivalent inputs. Equivalence is understood in the sense that the problem for them is solved in a similar way.

Keywords: *generic complexity, graph triangulation problem.*

[1]Работа поддержана грантом РНФ № 22-11-20019.

Введение

Проблема разбиения графа на треугольники состоит в следующем. Задан конечный простой граф с $3n$ вершинами. Необходимо определить, можно ли разбить вершины графа на n трёхэлементных множеств так, что в каждом из них все вершины соединены рёбрами исходного графа (то есть являются треугольниками). NP-полнота этой проблемы доказана Шейфером в 1974 г. и содержится в классической монографии [1]. При условии $\mathrm{P} \neq \mathrm{NP}$ не существует полиномиального алгоритма для решения этой задачи. Здесь P — это класс алгоритмических проблем распознавания, решаемых за полиномиально ограниченное время на детерминированных машинах Тьюринга, а класс NP состоит из проблем, решаемых за полиномиальное время на недетерминированных машинах Тьюринга.

Генерический подход [2] — 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 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. Отметим, что похожий подход для изучения проблем оптимизации предложен ранее в [3].

Большой интерес как с теоретической точки зрения, так и с точки зрения практических приложений представляют алгоритмические проблемы, которые остаются неразрешимыми или трудноразрешимыми и в генерическом случае. Например, 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 генерически легкоразрешима, то для почти всех таких входов её можно 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: распознавания квадратичных вычетов, дискретного логарифма, извлечения корня в группах вычетов (обращения функции RSA).

М. Блюм для доказательства того, что проблема дискретного логарифма является сложной для почти всех входов, при условии её трудноразрешимости в худшем случае, предложил метод амплификации [4]. Идея метода состоит в следующем. По заданному входу x алгоритмической проблемы A случайно генерируется «большое» множество входов $C(x)$, эквивалентных x в том смысле, что по решению проблемы для любого входа из $C(x)$ можно эффективно найти решение и для x. Если теперь допустить существование эффективного (полиномиального) генерического алгоритма для проблемы A, то из него можно получить вероятностный полиномиальный алгоритм, решающий проблему A для всех входов. Этот алгоритм по входу x путём генерации «большого» множества $C(x)$ как бы переносит вход x из области «плохих» входов для генерического алгоритма в область, где этот алгоритм может решить проблему. Таким образом, если проблема A трудноразрешима в худшем случае, то и генерически она тоже трудноразрешима. Оказалось, что метод амплификации применим и для перечисленных выше проблем криптографии.

Метод генерической амплификации был развит в [5]. С его помощью доказана генерическая неразрешимость и трудноразрешимость многих алгоритмических проблем:

проблема остановки для машин Тьюринга [6], проблема равенства в некоторых конечно определённых полугруппах [5], проблема разрешимости элементарных теорий, неразрешимых в классическом случае [5], арифметика Пресбургера [7], десятая проблема Гильберта [8], проблема выполнимости булевых формул [9], проблема кластеризации графов [10], проблема распознавания гамильтоновых графов [11].

Данная работа посвящена изучению генерической сложности разбиения графа на треугольники. Доказывается, что при условии $\mathrm{P} \neq \mathrm{NP}$ и $\mathrm{P} = \mathrm{BPP}$ для этой проблемы не существует полиномиального сильно генерического алгоритма. Сильно генерический алгоритм решает проблему не на всём множестве входов, а на подмножестве, последовательность частот которого при увеличении размера экспоненциально быстро сходится к 1. Класс BPP состоит из проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. Одной из важных гипотез в теории сложности вычислений является гипотеза о совпадении классов P и BPP. Из нее следует, что любой полиномиальный вероятностный алгоритм $\mathcal{A}$ можно эффективно дерандомизировать, то есть построить полиномиальный алгоритм $\mathcal{B}$, не использующий генератор случайных чисел и решающий ту же проблему, что и алгоритм $\mathcal{A}$. В работе [12] доказано, что равенство $\mathrm{P} = \mathrm{BPP}$ следует из весьма правдоподобных гипотез о вычислительной сложности некоторых трудных проблем.

1. Предварительные сведения

Пусть I — некоторое множество входов, I_n — подмножество входов размера n. Для подмножества $S \subseteq I$ определим последовательность

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

где $S_n = S \cap I_n$ — множество входов из S размера n. Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. *Асимптотической плотностью* S назовём предел

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

Верхний предел здесь нужен потому, что часто при кодировании входных данных не для каждого n существуют коды размера n. Множество S называется *пренебрежимым*, если $\rho(S) = 0$, и *сильно пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т. е. существуют константы σ, $0 < \sigma < 1$, и $C > 0$, такие, что для любого n имеет место $\rho_n(S) < C\sigma^n$.

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 $\{x \in I : \mathcal{A}(x) = ?\}$ является (сильно) пренебрежимым.

Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если для всех $x \in I$ выполнено

$$(\mathcal{A}(x) = y \in J) \Rightarrow (f(x) = y).$$

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). Различие между генерически разрешимыми проблемами и сильно генерически разрешимыми проблемами поясняется в работе [13].

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 $t_M(x) < p(|x|)$. Здесь через $|x|$ обозначен размер входа 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 или второе правило. 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 считают, 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 [12].

2. Разбиение графа на треугольники

Рассмотрим неориентированые графы без петель и кратных рёбер. *Проблема разбиения графа на треугольники* состоит в следующем. Пусть задан граф G с множеством вершин $V = \{v_1, \ldots, v_{3n}\}$. Необходимо определить, можно ли разбить вершины графа на n трёхэлементных множеств так, что в каждом из них все вершины соединены рёбрами исходного графа (то есть являются треугольниками). Набор из этих n трёхэлементных множеств будем называть *3-разбиением*.

Лемма 1. Пусть G_1 и G_2 — два графа с непересекающимися множествами вершин $V = \{v_1, \ldots, v_{3n}\}$ и $W = \{w_1, \ldots, w_{3m}\}$. Пусть граф G_2 можно разбить на треугольники. Тогда граф G_1 можно разбить на треугольники тогда и только тогда, когда граф $G_1 \cup G_2$ можно разбить на треугольники.

Доказательство. Пусть графы G_1 и G_2 можно разбить на треугольники и $P(G_1)$, $P(G_2)$ — соответствующие 3-разбиения. Тогда легко видеть, что 3-разбиение графа $G_1 \cup G_2$ есть $P(G_1) \cup P(G_2)$.

Обратно, пусть графы G_2 и $G_1 \cup G_2$ можно разбить на треугольники и $P(G_2)$, $P(G_1 \cup G_2)$ — соответствующие 3-разбиения. Так как подграфы G_1 и G_2 в графе $G_1 \cup G_2$ не имеют общих вершин и рёбер, 3-разбиением графа G_1 является $P(G_1 \cup G_2) \setminus P(G_2)$. ■

Будем использовать представление графов с помощью матриц смежности. Напомним, что матрица смежности $M(G)$ графа G с множеством вершин $v_1, \ldots, v_n$ — это матрица порядка n, в которой на месте (i, j) стоит 1, если в графе G есть ребро (v_i, v_j), и 0, если такого ребра нет. Так как графы неориентированные, для кодирования графа с n вершинами достаточно верхней части матрицы, состоящей из $n(n-1)/2$ бит. Таким образом, будем считать, что размер графа с n вершинами равен $n(n-1)/2$.

Теорема 1. Если P $\neq$ NP и P=BPP, то не существует сильно генерического полиномиального алгоритма для решения проблемы разбиения графа на треугольники.

Доказательство. Допустим, что существует сильно генерический полиномиальный алгоритм $\mathcal{A}$, решающий проблему разбиения графа на треугольники. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, решающий эту проблему на всём множестве входов. Это означает, что проблема разбиения графа на треугольники лежит в классе BPP. Так как P = BPP, она лежит и в классе P, откуда из NP-полноты данной проблемы следует P $\neq$ NP, что противоречит посылке теоремы.

Пусть имеется граф G с $3n$ вершинами $v_1, \ldots, v_{3n}$. Он имеет размер $3n(3n-1)/2$. Алгоритм $\mathcal{B}$ работает на графе G следующим образом:

1) Генерируем случайный граф H с $(3n)^2-3n$ вершинами $v_{3n+1}, \ldots, v_{(3n)^2}$, имеющий следующее 3-разбиение:

$$P(H) = \{(v_{3n+1}, v_{3n+2}, v_{3n+3}), \ldots, (v_{(3n)^2-2}, v_{(3n)^2-1}, v_{(3n)^2})\}.$$

   Это делаем таким образом. В верхней половине матрицы смежности графа H приравниваем 1 элементы, соответствующие рёбрам треугольников $(v_{3n+1}, v_{3n+2}, v_{3n+3})$, …, $(v_{(3n)^2-2}, v_{(3n)^2-1}, v_{(3n)^2})$. Этих фиксированных элементов будет $(3n)^2 - 3n$. Остальные $((3n)^2 - 3n)((3n)^2 - 3n - 1)/2 - (3n)^2 + 3n$ элементов выбираем из 0 и 1 случайно и равновероятно.
2) Запускаем алгоритм $\mathcal{A}$ на графе $G' = G \cup H$.
3) Если $\mathcal{A}(G') = 1$, то по лемме 1 в графе G есть 3-разбиение. Выдаём ответ «ДА».
4) Если $\mathcal{A}(G') = 0$, то по лемме 1 в графе G нет 3-разбиения. Выдаём ответ «НЕТ».
5) Если $\mathcal{A}(G') = ?$, то выдаём ответ «НЕТ».

Заметим, что полиномиальный вероятностный алгоритм $\mathcal{B}$ выдаёт правильный ответ на шагах 3 и 4, а на шаге 5 может выдать неправильный ответ. Нужно доказать, что вероятность того, что ответ выдаётся на шаге 5, меньше 1/3.

Граф $G \cup H$ имеет $(3n)^2$ вершин, то есть его размер равен $m = ((3n)^4 - (3n)^2)/2$. Вероятность того, что для случайного графа $G \cup H$ имеет место $\mathcal{A}(G \cup H) = ?$, не больше

$$\frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|} = \frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\mathcal{G}_m|} \cdot \frac{|\mathcal{G}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|}.$$

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

$$\frac{|\{G \in \mathcal{G} : \mathcal{A}(G) \neq ?\}_m|}{|\mathcal{G}_m|} < \frac{1}{2^{\alpha m}} = \frac{1}{2^{\alpha((3n)^4-(3n)^2)/2}} = \frac{1}{2^{\alpha(81n^4-9n^2)/2}}$$

для любого n.

С другой стороны, так как матрица смежности графа H имеет $((3n)^2 - 3n)((3n)^2 - - 3n - 1)/2 - (3n)^2 + 3n$ элементов, которые можно выбирать произвольно, то

$$|\{G \cup H : H \in \mathcal{G}\}_m| = 2^{((3n)^2-3n)((3n)^2-3n-1)/2-(3n)^2+3n} = 2^{(81n^4-54n^3-18n^2+9n)/2}.$$

Отсюда

$$\frac{|\mathcal{G}_m|}{|\{G \cup H : H \in \mathcal{G}\}_m|} = \frac{2^{(81n^4-9n^2)/2}}{2^{(81n^4-54n^3-18n^2+9n)/2}} = 2^{(54n^3+9n^2-9n)/2}.$$

Поэтому искомая вероятность не больше

$$\frac{2^{(54n^3+9n^2-9n)/2}}{2^{\alpha(81n^4-9n^2)/2}} < \frac{1}{3}$$

при достаточно больших n. ■

Автор выражает благодарность рецензенту за полезные замечания и предложения по улучшению текста статьи.

ЛИТЕРАТУРА

1. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 420 с.
2. *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.
3. *Гимади Э. Х., Глебов Н. И., Перепелица В. А.* Алгоритмы с оценками для задач дискретной оптимизации // Проблемы кибернетики. 1975. Т. 31. С. 35–42.
4. *Blum M.* How to prove a theorem so no one else can claim it // Proc. Intern. Congress Math., Berkeley, CA, 1986. P. 1444–1451.
5. *Myasnikov A. G. and Rybalov A. N.* Generic complexity of undecidable problems // J. Symbolic Logic. 2008. V. 73. No. 2. P. 656–673.
6. *Rybalov A. N.* On the strongly generic undecidability of the Halting Problem // Theor. Comput. Sci. 2007. V. 377. P. 268–270.
7. *Rybalov A. N.* Generic complexity of Presburger arithmetic // Theory Comput. Systems. 2010. V. 46. No. 1. P. 2–8.
8. *Rybalov A. N.* Generic complexity of the Diophantine problem // Groups Complexity Cryptology. 2013. V. 5. No. 1. P. 25–30.
9. *Rybalov A. N.* Generic hardness of the Boolean satisfiability problem // Groups Complexity Cryptology. 2017. V. 9. No. 2. P. 151–154.
10. *Рыбалов А. Н.* О генерической сложности проблемы кластеризации графов // Прикладная дискретная математика. 2019. № 46. С. 72–77.
11. *Рыбалов А. Н.* О генерической сложности проблемы распознавания гамильтоновых путей // Прикладная дискретная математика. 2021. № 53. С. 120–126.
12. *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.
13. *Рыбалов А.* О генерической сложности проблемы общезначимости булевых формул // Прикладная дискретная математика. 2016. № 2(32). С. 119–126.

REFERENCES

1. *Garey M. and Johnson D.* Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, 1979. 340 p.
2. *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.
3. *Gimadi E. H., Glebov N. I., and Perepelitsa V. A.* Algoritmy s otsenkami dlya zadach diskretnoy optimizatsii [Algorithms with bounds for problems of discrete optimization]. Problemy Kibernetiki, 1975, vol. 31, pp. 35–42. (in Russian)

4. *Blum M.* How to prove a theorem so no one else can claim it. Proc. Intern. Congress Math., Berkeley, CA, 1986, pp. 1444–1451.
5. *Myasnikov A. G. and Rybalov A. N.* Generic complexity of undecidable problems. J. Symbolic Logic, 2008, vol. 73, no. 2, pp. 656–673.
6. *Rybalov A. N.* On the strongly generic undecidability of the Halting Problem. Theor. Comput. Sci., 2007, vol. 377, pp. 268–270.
7. *Rybalov A. N.* Generic complexity of Presburger arithmetic. Theory Comput. Systems, 2010, vol. 46, no. 1, pp. 2–8.
8. *Rybalov A. N.* Generic complexity of the Diophantine problem. Groups Complexity Cryptology, 2013, vol. 5, no. 1, pp. 25–30.
9. *Rybalov A. N.* Generic hardness of the Boolean satisfiability problem. Groups Complexity Cryptology, 2017, vol. 9, no. 2, pp. 151–154.
10. *Rybalov A. N.* O genericheskoy slozhnosti problemy klasterizatsii grafov [On generic complexity of the graph clustering problem]. Prikladnaya Diskretnaya Matematika, 2019, no. 46, pp. 72–77. (in Russian)
11. *Rybalov A. N.* O genericheskoy slozhnosti problemy raspoznavaniya gamil'tonovykh putey [The general complexity of the problem to recognize Hamiltonian paths]. Prikladnaya Diskretnaya Matematika, 2021, no. 53, pp. 120–126. (in Russian)
12. *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.
13. *Rybalov A. N.* O genericheskoy slozhnosti problemy obshcheznachimosti bulevykh formul [On generic complexity of the validity problem for Boolean formulas]. Prikladnaya Diskretnaya Matematika, 2016, no. 2(32), pp. 119–126. (in Russian)