

УДК 519.681.4

ПРИБЛИЖЕННОЕ ВЫЧИСЛЕНИЕ ВЕСОВОЙ ФУНКЦИИ ЛИНЕЙНОГО ДВОИЧНОГО КОДА^{*)}

М. Н. Вялый

Рассматривается задача приближенного вычисления весовой функции двоичного линейного кода в точках единичной окружности с аддитивной погрешностью ε . Приближения весовых функций линейных двоичных кодов с аддитивной погрешностью возникают при анализе квантовых алгоритмов. Задача точного вычисления значения весовой функции вычислительно трудна. В работе показано, что приближенного вычисления весовой функции с аддитивной погрешностью 2^{n-n^c} , где $c < 1$ — положительная константа, достаточно для точного определения коэффициентов весовой функции. Аналогичные, но более слабые, результаты получены также для задачи приближенного вычисления весовой функции в единственной точке (основной пример — в точке $\omega = e^{i\pi/4}$).

1. Формулировки задач и основных результатов

Всюду далее линейный двоичный код для краткости называется кодом. $[n, k]$ -код — это код длины n и размерности k .

Под весовой функцией кода C будем понимать следующий многочлен

$$w_C(y) = \sum_{\mathbf{x} \in C} y^{|\mathbf{x}|} = \sum_{j=0}^n w_j(C) y^j.$$

В последнем равенстве w_j обозначает число кодовых слов веса j , а n — длину кода.

В данной статье рассматривается алгоритмическая сложность задачи вычисления (точного или приближенного) весовой функции кода. По весовой функции легко определить минимальное расстояние кода, тогда как с вычислительной точки зрения задача вычисления весовой функции

^{*)}Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проекты 02-01-00547 и 02-01-22001 НЦНИ_а) и государственной программы поддержки ведущих научных школ РФ (проект НШ 1721.2003.1.)

является сложной. Как показано в [12], задача определения минимального расстояния кода NP-трудна. Более того, даже приближенное вычисление минимального расстояния оказывается NP-трудной задачей [7].

Известно несколько формулировок алгоритмических задач, связанных с вычислением весовой функции.

Задача CF: вычисление коэффициентов. Входом задачи является код C , выходом — набор коэффициентов весовой функции $w_C(y)$.

Здесь и всюду в дальнейшем мы подразумеваем эффективное задание кода, например, порождающей или проверочной матрицей.

В задачах вычисления значения весовой функции входом является описание кода, указание аргумента, от которого вычисляется весовая функция, и требуемой точности результата. Наиболее общий вид задачи вычисления весовой функции таков.

Задача WF Входом задачи является тройка (код C , аргумент y , точность ε), выходом — такое число \tilde{w} , что $|\tilde{w} - w_C(y)| < \varepsilon$.

Аргументы y, ε в этой задаче можно считать эффективно вычислимыми числами. Действительное число α называется *эффективно вычислимым*, если существует такой алгоритм, который по входу δ (конечная двоичная дробь) за время, полиномиальное от длины записи δ , выдает такую конечную двоичную дробь $\alpha(\delta)$, что $|\alpha(\delta) - \alpha| < \delta$. Комплексное число называется эффективно вычислимым, если эффективно вычислимы его вещественная и мнимая части.

Далее всюду предполагается, что все параметры и входные данные задач, которые являются действительными или комплексными числами, эффективно вычислимы.

Впрочем, дальнейшие результаты и рассуждения не меняются, если считать, что вещественная и мнимая части числа y , равно как и точность ε , заданы в виде конечных двоичных дробей.

Фиксируя аргумент весовой функции и/или заданную точность, получаем различные специализации задачи WF. Алгоритмическая сложность таких специализированных задач может быть, вообще говоря, ниже, чем у задачи WF.

Задача WF(y) Входом задачи является пара (код C , точность ε), выходом — такое число \tilde{w} , что $|\tilde{w} - w_C(y)| < \varepsilon$.

Это — целая серия задач, параметризованных по y . Хотя определение осмысленно для произвольного y , мы будем рассматривать задачу WF(y) только для эффективно вычисляемых y .

Нас интересуют значения весовой функции в точках единичной окруж-

ности (при $|y| = 1$). Если $|y| = 1$, то $|w_C(y)| \leq 2^k \leq 2^n$ для $[n, k]$ -кода C . Это отчасти оправдывает выбор порога точности в приводимых ниже формулировках задач приближенного вычисления весовой функции.

Задача WE(α): приближенное вычисление весовой функции. Входом задачи является пара (код C , аргумент y), выходом — такое число \tilde{w} , что $|\tilde{w} - w_C(y)| < 2^{\alpha n}$.

Задача WE(α, y): приближенное вычисление весовой функции от фиксированного аргумента y . Входом задачи является код C , выходом — такое число \tilde{w} , что $|\tilde{w} - w_C(y)| < 2^{\alpha n}$.

Всюду далее рассматриваются только задачи WE(α) и WE(α, y) при $0 < \alpha < 1$.

Забегая вперед, отметим, что интерес к такой постановке (аддитивная погрешность, точность порядка $2^{\alpha n}$) связан с анализом вычислительных возможностей квантовой модели вычислений. Оказывается (см. ниже следствие 2), что класс BQP задач, для которых существуют эффективные квантовые алгоритмы (определение см. ниже в разделе 2), сводится к задаче WE($1/2, e^{i\pi/4}$). Говоря другими словами, задача WE($1/2, e^{i\pi/4}$) BQP-трудна. Более того, как показывает теорема 3 (см. ниже), эта задача полна в предположительно более широком классе $P^{\#P}$, так что ее принадлежность классу BQP сомнительна.

Стандартные классы в теории сложности являются классами предикатов (задач распознавания). Поэтому алгоритмическую сложность рассматриваемых здесь задач будем описывать сложностным классом P^A предикатов, распознаваемых за полиномиальное время детерминированными машинами Тьюринга, которые снабжены оракулом, решающим задачу A . При таком подходе естественно рассматривать полноту и сводимость по Тьюрингу (оракульную сводимость) за полиномиальное время, что и делается в дальнейшем.

Рассматриваемые задачи мы будем сравнивать с классом $P^{\#P}$. Функция $f: \mathbb{N} \rightarrow \mathbb{N}$ принадлежит классу $\#P$, если существует двуместный предикат $R(x, y)$, распознаваемый за полиномиальное время, и многочлен $p(n)$ такие, что

$$f(x) = |\{y \mid |y| = p(|x|) \text{ и } R(x, y)\}|.$$

Класс $P^{\#P}$ состоит из предикатов, распознаваемых за полиномиальное время машинами Тьюринга с оракулом из $P^{\#P}$. Как следует из теоремы Тода [11], класс $P^{\#P}$ содержит все классы полиномиальной иерархии (в частности, класс NP) и класс RP. Последний можно определить так:

предикат $L(x)$ принадлежит RP , если существуют две $\#P$ -функции f, g такие, что $L(x) = 1$ тогда и только тогда, когда $f(x) > g(x)$.

Включение $BQP \subseteq RP$ близко к наиболее точной верхней оценке возможностей эффективного квантового вычисления в терминах классических сложностных классов. Наиболее точная оценка и теоретико-сложностные аргументы в пользу гипотезы $BQP \subset RP$ содержатся в работе [8].

Для вычисления весовой функции достаточно знать ее коэффициенты. Задача вычисления коэффициентов принадлежит классу $P^{\#P}$. Поэтому и все задачи, связанные с вычислением весовой функции, также принадлежат классу $P^{\#P}$. Многие из них оказываются полными в этом классе. В некоторых исключительных случаях значение весовой функции может быть найдено за полиномиальное время.

Для вычислений с неограниченной точностью справедливы следующие утверждения.

Теорема 1. $P^{CF} = P^{WF} = P^{WF(y)} = P^{\#P}$ для y вида $e^{i\pi t/s}$, $s = 4p$, p — целое, а t взаимно просто с $8p + 1$.

Теорема 2. Задачи $WF(1)$, $WF(-1)$, $WF(i)$, $WF(-i)$ решаются за полиномиальное время.

Доказательство теоремы 2 приводится в разделе 3.

Для вычислений с ограниченной точностью имеется следующий набор полных задач.

Теорема 3. $P^{WE(\alpha)} = P^{\#P}$ при $0 < \alpha < 1$, $P^{WE(\alpha, \omega_p^t)} = P^{\#P}$, где $p \geq 1$ — целое, t взаимно просто с $8p + 1$, $\omega_p = e^{i\pi/4p}$ и $0 < \alpha < \log_2 |1 + \omega_p|$.

Доказательства теорем 1 и 3 основаны на ряде сведений, которые мы сформулируем в виде лемм. Непосредственно из определений получаем следующие леммы.

Лемма 1. Задача $WF(y)$ сводится к задаче WF .

Лемма 2. Задача $WE(\alpha, y)$ сводится к задаче $WE(\alpha)$.

Если известны коэффициенты многочлена, то за полиномиальное время можно найти его значение от эффективно вычислимого аргумента с экспоненциально малой погрешностью. Из этого соображения получаем следующие сведения.

Лемма 3. Задача WF сводится к задаче CF .

Лемма 4. Задача $WE(\alpha)$ сводится к задаче CF .

Лемма 5. Задача CF принадлежит классу $\#P$.

Доказательство. Проверка принадлежности слова коду и подсчет числа единиц в слове могут быть осуществлены за полиномиальное время. Поэтому функция $(j, C) \mapsto w_j(C)$, вычисляющая коэффициенты весовой функции кода, принадлежит классу $\#P$.

Из сформулированных выше лемм следует, что для всех рассматриваемых здесь задач A выполняется включение $P^A \subseteq P^{\#P}$.

Лемма 6. Задача $WF(y^r)$ сводится к задаче $WF(y)$ при любом натуральном r .

Доказательство. Пусть rC — код, полученный r -кратным повторением каждого кодового слова кода C . Тогда $w_{rj}(rC) = w_j(C)$, а остальные коэффициенты $w_j(rC)$ равны 0. Поэтому

$$w_C(y^r) = w_{rC}(y).$$

Лемма 7. Задача $\#P$ сводится к задаче $WF(\omega)$, где $\omega = e^{i\pi/4}$.

Доказательство леммы 7 использует еще одно промежуточное сведение к задаче оценки матричного элемента полилокального унитарного оператора и теорему Китаева–Соловея, позволяющую эффективно строить приближения унитарных операторов. Эта часть доказательства и необходимые сведения из теории квантовых вычислений описаны в следующем разделе, а завершение доказательства леммы 7 приводится в разделе 3.

Лемма 8. Задача CF сводится к $WE(\alpha)$ при $0 < \alpha < 1$.

Лемма 9. Задача $WF(\omega_p)$ сводится к задаче $WE(\alpha, \omega_p)$ при $0 < \alpha < \log_2 |1 + \omega_p|$.

Доказательства этих лемм основаны на том, что задача приближенного вычисления значения весовой функции обладает свойством самосводимости с увеличением точности. Они приводятся в разделе 4 (наряду с усиленным вариантом леммы 8): оказывается, что точность вычисления весовой функции вида 2^{n-n^c} , $c < 1$, достаточна для точного определения коэффициентов весовой функции.

Леммы 7–9 дают примеры полных задач вычисления весовой функции. Для завершения доказательства теорем 1 и 3 их нужно скомбинировать с леммой 6.

2. Теорема Китаева–Соловея и ее применение к построению полных задач

Через $SU(n)$ обозначим специальную группу унитарных операторов порядка n (определитель равен 1).

Теорема 4. Пусть операторы X_1, \dots, X_l из $SU(n)$ порождают всюду плотное множество в $SU(n)$, $n \geq 3$. Тогда существует алгоритм, который по оператору $U \in SU(n)$ и заданной точности δ строит δ -приближение \tilde{U} оператора U в виде произведения операторов X_1, \dots, X_l и обратных к ним операторов $X_1^{-1}, \dots, X_l^{-1}$. Временная сложность алгоритма равна $\exp(O(n)) \text{poly} \log(1/\delta)$.

Расстояние между операторами измеряется в операторной норме. Константа в оценке времени работы зависит от множества образующих X_j (и может быть сколь угодно велика). Унитарные операторы представляются матрицами, матричные элементы которых являются эффективно вычислимыми комплексными числами.

Краткий набросок доказательства теоремы Китаева–Соловея можно найти в [2]. Подробное доказательство содержится в английском переводе книги [3].

Эта теорема применяется в теории квантовых вычислений для моделирования квантовых схем, заданных в некотором базисе операторов, квантовыми схемами в полном базисе операторов. Напомним основные определения, относящиеся к квантовым схемам; более подробное изложение можно найти в [3].

Здесь и далее мы считаем, что в пространстве \mathbb{C}^2 зафиксировано скалярное произведение $\langle x, y \rangle$ и ортонормированный базис e_0, e_1 , а в пространстве $(\mathbb{C}^2)^{\otimes n}$ — базис из тензорных произведений e_j (*вычислительный базис*). Элементы этого базиса естественно индексируются двоичными словами длины n :

$$e_x = e_{x_1} \otimes e_{x_2} \otimes \dots \otimes e_{x_n},$$

а матричные элементы операторов — парами двоичных слов. В соответствии с этим соглашением $U_{0^n, 0^n}$ — матричный элемент оператора U в вычислительном базисе. Скалярное произведение в пространстве $(\mathbb{C}^2)^{\otimes n}$ будем также обозначать угловыми скобками \langle, \rangle .

Квантовая схема — это представление унитарного оператора U , действующего в пространстве $(\mathbb{C}^2)^{\otimes n}$, тензорные сомножители которого называются кубитами. Оператор представляется в виде произведения

$$U = U_L[A_L] \dots U_1[A_1],$$

в котором каждый сомножитель U_j действует нетривиально только на кубиты из подмножества $A_j \subset \{1, \dots, n\}$ (это подмножество называется квантовым регистром). Операторы U_j берутся из некоторого множества

\mathcal{B} , называемого *базисом операторов*. Число L называется *размером* схемы.

Базис операторов называется *полным*, если он замкнут относительно взятия обратного оператора, а его элементы порождают всюду плотное множество в $U((\mathbb{C}^2)^{\otimes r})/U$, $r \geq 2$; здесь фактор берется по группе скалярных операторов.

Примером полного базиса операторов является базис

$$\mathcal{T} = \{\text{CNOT}, H, T\}, \quad \text{где}$$

$$\text{CNOT}: e_{a,b} \mapsto e_{a,a \oplus b}; \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \quad (1)$$

(см. [6,10]). Оператор CNOT называется еще оператором обратимого копирования бита, оператор H — оператором Адамара, а оператор T — оператором фазового сдвига (на $\pi/8$). Заметим, что для формального соответствия определению полного базиса нужно еще добавить T^{-1} . Но $T^{-1} = T^7$, а нас интересуют только оценки размера с точностью до мультипликативной константы и даже грубее.

Используя любой конечный полный базис операторов, скажем \mathcal{T} , можно определить класс эффективных квантовых вычислений BQP. Прежде всего определим вероятность успеха $\text{Acc}(U)$ для схемы U :

$$\text{Acc}(U) = |\Pi^1[1]Ue_{0^n}|^2, \quad \Pi^1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

(запись $\Pi^1[1]$ означает, что оператор Π^1 действует на первый кубит.) Далее заметим, что описание квантовой схемы в конечном базисе \mathcal{T} — это указание последовательности базисных операторов и регистров, к которым они применяются. Предикат $L(x)$ принадлежит BQP, если существует алгоритм для машины Тьюринга, который за время, полиномиальное от длины записи x , строит описание квантовой схемы $U(x)$ в базисе \mathcal{T} , причем выполняются условия:

$$\begin{aligned} &\text{если } L(x), \quad \text{то } \text{Acc}(U(x)) > 1 - 1/3, \\ &\text{если не } L(x), \text{ то } \text{Acc}(U(x)) < 1/3. \end{aligned}$$

Не вдаваясь в подробные комментарии этого определения, отметим, что порог ошибки $1/3$ можно сделать экспоненциально малым, а базис \mathcal{T} заменить на любой полный базис.

Теперь рассмотрим два базиса операторов \mathcal{B} и \mathcal{B}' . Предположим, что \mathcal{B} — полный, а \mathcal{B}' состоит из операторов, которые действуют на регистры ограниченного размера. Такие операторы называются локальными.

Из теоремы Китаева–Соловея следует, что оператор, заданный схемой полиномиального размера в базисе \mathcal{B}' , можно эффективно приблизить схемой полиномиального размера в базисе \mathcal{B} с экспоненциально малой точностью. Для этого нужно строить точные приближения каждого сомножителя схемы. В силу локальности экспоненциальный множитель в оценке времени работы алгоритма из теоремы Китаева–Соловея становится константой. Общая оценка точности приближения увеличивается не слишком сильно: благодаря унитарности операторов ошибки всего лишь складываются.

Это следствие можно использовать для построения задач, полных (по Тьюрингу) в классе $P^{\#P}$. Они получаются сведениями из задачи оценки матричного элемента.

Пусть \mathcal{B} — некоторый базис операторов. Задача оценки матричного элемента для операторов в базисе \mathcal{B} определяется следующим образом.

Задача $ME(\mathcal{B})$. Входом задачи является пара (схема вычисления оператора U в базисе \mathcal{B} , точность ε), выходом — такое число \tilde{u} , что $|\tilde{u} - U_{0^n, 0^n}| < \varepsilon$.

Рассмотрим базис операторов $\mathcal{S} = \{S^r, H\}$, $0 \leq r \leq 3$, где H — оператор Адамара, а $S^r: (\mathbb{C}^2)^{\otimes r} \rightarrow (\mathbb{C}^2)^{\otimes r}$ — оператор условного фазового сдвига с r управляющими кубитами:

$$\begin{cases} S^r e_{x_1, \dots, x_r} = -e_{x_1, \dots, x_r}, & \text{если } x_1 = x_2 = \dots = x_r = 1, \\ S^r e_{x_1, \dots, x_r} = e_{x_1, \dots, x_r} & \text{в противном случае.} \end{cases}$$

Лемма 10. Любая функция f из класса $\#P$ может быть вычислена за полиномиальное время на машине Тьюринга с оракулом $ME(\mathcal{S})$.

Доказательство. Достаточно рассмотреть одну $\#P$ -полную функцию f . В качестве такой функции возьмем функцию, которая булевой схеме в базисе из элементов $\{\oplus, \wedge, 1\}$ сопоставляет число выполняющих наборов переменных, т. е. наборов, на которых эта схема дает 1. Поскольку базис $\{\oplus, \wedge, 1\}$ полный, а КНФ является частным случаем булевой схемы (и даже формулы) в стандартном базисе $\{\vee, \wedge, \neg\}$, имеем очевидную сводимость стандартной $\#P$ -полной задачи подсчета выполняющих наборов для КНФ [1] к задаче подсчета числа выполняющих наборов для булевой схемы в базисе $\{\oplus, \wedge, 1\}$.

Итак, пусть дана схема C в базисе $\{\oplus, \wedge, 1\}$. Через n обозначим число входных переменных схемы C , через s — размер схемы C , через z_j , $1 \leq j \leq s$, — ее вспомогательные переменные, через $\#(C)$ — число выполняющих наборов. Считаем также, что выходная переменная схемы —

z_s . Каждое присваивание в схеме имеет вид $z_j := a * b$, где $*$ $\in \{\oplus, \wedge\}$, а a, b — либо входные переменные, либо вспомогательные переменные с меньшим индексом, либо константа 1. Положим $Z_j = z_j + a * b$ и рассмотрим систему уравнений от $n + s$ переменных

$$\begin{cases} Z_j(x, z) = 0, & 1 \leq j \leq s, \\ z_s = 1. \end{cases} \quad (2)$$

Так как значения всех вспомогательных переменных схемы однозначно определяются значениями входных переменных, то число выполняющих наборов переменных для схемы C совпадает с числом решений системы (2).

Для полинома g над полем из двух элементов \mathbb{F}_2 введем обозначение

$$\Delta g = \sum_x (-1)^{g(x)}. \quad (3)$$

Теперь построим полином над \mathbb{F}_2 от $N = n + 2s + 1$ переменных x_1, \dots, x_n (входные переменные схемы), z_1, \dots, z_s (вспомогательные переменные схемы), v_0, v_1, \dots, v_s :

$$F_C(x, z, v) = \sum_{j=1}^s v_j Z_j(x, z) + v_0(1 + z_s). \quad (4)$$

Докажем, что

$$\Delta F_C(x, z, v) = 2^{s+1} \#(C). \quad (5)$$

Из формулы (4) видно, что F_C — линейный полином по переменным v_j . Поэтому наборы значений переменных x, z , не удовлетворяющие системе (2), дают нулевой вклад в сумму (3) для полинома F_C . Если же набор x^0, z^0 является решением системы (2), то $F_C(x^0, z^0, v) = 0$ и такой набор дает вклад 2^{s+1} в сумму (3).

В силу (5) вычисление $\#(C)$ свелось к вычислению ΔF_C . Для простоты записи формул переобозначим переменные полинома F_C через u_1, u_2, \dots, u_N .

Теперь зададим квантовую схему для унитарного оператора $U(F_C): (\mathbb{C}^2)^{\otimes N} \rightarrow (\mathbb{C}^2)^{\otimes N}$ в базисе \mathcal{S} .

Запишем полином F_C в виде суммы мономов. Это можно сделать за полиномиальное время, поскольку степень F_C равна 3, так что число ненулевых мономов в F_C не превосходит $\binom{N}{3}$. Каждому моному $u_J = \prod_{j \in J} u_j$ поставим в соответствие оператор $S_J = S^{|J|}[J]$.

Искомая квантовая схема имеет вид

$$U(F_C) = \prod_{j=1}^N H[j] \prod_{J \in M(F_C)} S_J \prod_{j=1}^N H[j] = H^{\otimes N} S(F_C) H^{\otimes N}, \quad (6)$$

где $M(F_C)$ — множество мономов полинома F_C .

Положим $\varepsilon = 2^{-N-2}$. Используя оракул $ME(S)$, найдем ε -приближение U_0 матричного элемента $U(F_C)_{0^N, 0^N}$. Ближайшее к $2^N U_0$ целое число равно ΔF_C , как следует из вычисления матричного элемента $U(F_C)_{0^N, 0^N}$. Оператор $H^{\otimes N}$ унитарный, а его квадрат равен тождественному оператору. Поэтому $U(F_C)_{0^N, 0^N} = \langle H^{\otimes N} e_{0^N}, S(F_C) H^{\otimes N} e_{0^N} \rangle$. Так как

$$H^{\otimes N} e_{0^N} = \frac{1}{2^{N/2}} \sum_x e_x,$$

получаем

$$\begin{aligned} 2^N U(F_C)_{0^N, 0^N} &= \sum_{u, v} \langle e_u, S(F_C) e_v \rangle = \sum_u (-1)^{\sum_{J \in M(F_C)} u_J} \langle e_u, e_u \rangle = \\ &= \sum_u (-1)^{F_C(u)} = \Delta F_C. \end{aligned}$$

Лемма 10 доказана.

Следствие 1. Любая функция f из класса $\#P$ может быть вычислена за полиномиальное время на машине Тьюринга с оракулом $ME(\mathcal{B})$ для любого конечного полного базиса \mathcal{B} .

Доказательство. Заметим, что схема (6) состоит из локальных операторов (каждый действует не более чем на 3 бита). Применяя алгоритм из теоремы Китаева–Соловея, за полиномиальное время найдем приближения в базисе \mathcal{B} каждого множителя в (6) с точностью $\delta = 2^{-N-2}/M$, где M — размер схемы (6). Обозначим через \tilde{U} произведение этих приближений. Из-за унитарности операторов ошибка накапливается линейно, так что \tilde{U} дает хорошее приближение к $U(F_C)$:

$$|(\tilde{U} - U(F_C))_{0^N, 0^N}| < 2^{-N-2}.$$

3. Точное вычисление весовой функции

Доказательство теоремы 2. Для любого $[n, k]$ -кода имеем $w_C(1) = 2^k$. С учетом леммы 6 осталось построить полиномиальный алгоритм вычисления $w_C(i)$.

Без ограничения общности считаем, что код C задан порождающей матрицей полного ранга. В такой матрице имеется k строк, которые образуют базис для кода C , а ее столбцы задают коэффициенты в этом базисе линейных форм $c_j(u)$, $u \in \{0, 1\}^k$, выражающих значение j -го бита кодового слова.

Имеем

$$w_C(i) = \sum_u \prod_{j=1}^n i^{c_j(u)}. \quad (7)$$

Последовательно применяя равенство $i^a i^b = i^{a \oplus b} (-1)^{ab}$, можно переписать (7) в виде

$$w_C(i) = \sum_u i^{c(u)} (-1)^{Q(u)}, \quad \text{где } c(u) = \bigoplus_{j=1}^n c_j(u),$$

где $Q(u)$ — некоторая квадратичная форма. Задача вычисления весовой функции кода свелась к двум задачам вычисления выражений вида $\Delta Q_j(u)$ для полиномов второй степени не более чем от n переменных (Q_j — это ограничение Q на аффинное подпространство $c(u) = j$).

Такую задачу легко решить, приведя полином второй степени к каноническому виду

$$x_1 x_2 + x_3 x_4 + \dots + x_{2t-1} x_{2t} + x_{2t+1} + \dots + x_{2t+s}$$

(теорема Диксона [4, 5], доказательство конструктивно и автоматически превращается в алгоритм построения канонического вида за полиномиальное время). Легко видеть, что выражение Δf не меняется при аффинных заменах координат. Для канонического вида полинома от m переменных Δf вычисляется явно: $\Delta f = 0$ при $s > 0$ и $\Delta f = 2^{m-t}$ при $s = 0$.

Доказательство леммы 7. Построим алгоритм, который решает задачу $\text{ME}(\mathcal{T})$ за полиномиальное время с использованием оракула для задачи $\text{WF}(\omega)$, $\omega = e^{i\pi/4}$. Здесь \mathcal{T} — базис операторов (1).

Пусть дана квантовая схема в базисе \mathcal{T} , которая реализует оператор U , действующий на пространство n кубитов:

$$U = U_L U_{L-1} \dots U_1.$$

Матричный элемент $U_{0^n, 0^n}$ представим в виде суммы весов по путям вычисления:

$$U_{0^n, 0^n} = \sum_{x_1, x_2, \dots, x_{L-1}} (U_L)_{0^n, x_{L-1}} (U_{L-1})_{x_{L-1}, x_{L-2}} \dots (U_1)_{x_1, 0^n}, \quad (8)$$

где $x_j \in \{0, 1\}^n$. Все ненулевые веса имеют одну и ту же абсолютную величину $1/2^{h/2}$, где h — число операторов Адамара в схеме. Заметим, что операторы Адамара, которые применяются последними к некоторому кубиту, не влияют на число путей вычисления с ненулевыми весами. Обозначим через k число остальных операторов Адамара. Сумма (8) состоит из 2^k слагаемых. Эти слагаемые пометим двоичными наборами $u \in \{0, 1\}^k$, в которых записано значение кубита непосредственно после применения соответствующего оператора Адамара (по этим данным однозначно восстанавливается весь путь вычисления).

Фазовый сдвиг вдоль пути вычисления состоит из двух множителей. Первый множитель отвечает действию операторов T и имеет вид $\prod_{j=1}^t \omega^{\ell_j(u)}$, где $\ell_j(u)$ есть \mathbb{F}_2 -линейная форма. Второй множитель отвечает применению операторов Адамара. Его можно записать в виде $(-1)^{B(u)}$, где $B(u) = \sum_{j=1}^k b_j(u)u_j$. В этом выражении b_j есть \mathbb{F}_2 -линейная форма, ее значение равно значению кубита перед применением j -го оператора Адамара. Используя соотношения $i^{a \oplus b} = i^a i^b (-1)^{ab}$ и $i = \omega^2$, второй множитель можно записать в виде, аналогичном первому сомножителю. Получаем следующее выражение для матричного элемента:

$$U_{0^n, 0^n} = \frac{1}{2^{h/2}} \sum_{u \in \{0, 1\}^k} \prod_{j=1}^m \omega^{\beta_j(u)}, \quad (9)$$

где β_j суть \mathbb{F}_2 -линейные формы. Заметим, что указанные выше преобразования можно сделать эффективно, так что коэффициенты этих форм можно найти за полиномиальное время.

С точностью до множителя $1/2^{h/2}$ правая часть (9) совпадает с $w_C(\omega)$ для кода C , столбцы порождающей матрицы которого совпадают с коэффициентами линейных форм $\beta_j(u)$.

Таким образом, с точностью ε значение $U_{0^n, 0^n}$ получается делением $w_C(\omega)$ на $2^{h/2}$ с точностью $2^{h/2}\varepsilon$.

Следствие 2. Класс BQR сводится к задаче $WE(1/2, e^{i\pi/4})$.

Доказательство. Из работы [9] следует, что класс BQR сводится к следующей задаче: найти знак матричного элемента $U_{0^n, 0^n}$ оператора, заданного квантовой схемой в полном базисе, при условии, что $|\operatorname{Re} U_{0^n, 0^n}|$ отличается от 1 не более чем на фиксированную сколь угодно малую константу. Ясно, что описанного в доказательстве леммы 7 сведения достаточно для решения этой задачи. Поскольку $h > d$, в качестве константы

можно взять любое число $\delta < 1 - 1/\sqrt{2}$.

4. Приближенное вычисление весовой функции

При сведении задачи нахождения коэффициентов весовой функции к задаче приближенного вычисления весовой функции мы будем использовать операцию *прямой суммы* кодов.

Рассмотрим $[n_1, k_1]$ -код A и $[n_2, k_2]$ -код B . Обозначим через G_A и G_B порождающие матрицы для кодов A и B соответственно. Прямая сумма $A \oplus B$ — это $[n_1 + n_2, k_1 + k_2]$ -код, задаваемый порождающей матрицей вида

$$\begin{pmatrix} G_A & 0 \\ 0 & G_B \end{pmatrix}.$$

Весовая функция прямой суммы является произведением весовых функций слагаемых:

$$w_{A \oplus B}(y) = w_A(y)w_B(y). \quad (10)$$

Доказательство леммы 8. Для нахождения коэффициентов весовой функции $[n, k]$ -кода C мы будем использовать приближенные значения весовой функции кода $C_t = C \oplus I_t$ от аргументов y_j в окрестности 1. Значение параметра t будет указано позднее. I_t обозначает тривиальный код с единичной порождающей матрицей.

Выберем такое r , при котором справедливо неравенство $\beta = 2^\alpha/(2 - r) < 1$. Из приближения с точностью $2^{\alpha(n+t)}$ для w_{C_t} можно получить гораздо более точное приближение для w_C . Пусть $|y - 1| < r$ и $|w_{C_t}(y) - \tilde{w}| < 2^{\alpha(n+t)}$. Тогда с учетом (10) имеем

$$\left| \frac{w_C(y)(1+y)^t - \tilde{w}}{(1+y)^t} \right| = \left| w_C(y) - \frac{\tilde{w}}{(1+y)^t} \right| < 2^{\alpha n} \beta^t. \quad (11)$$

Последнее выражение при $t \rightarrow \infty$ убывает как $2^{-\Omega(t)}$.

Для вычисления коэффициентов весовой функции $w_C(y)$ выберем $n + 1$ точку круга $|z - 1| < r$ и вычислим с указанной в (11) точностью значения весовой функции от соответствующих аргументов. Расстояния между точками можно выбрать большими чем $r/(2(n + 1))$. Из интерполяционной формулы Лагранжа видно, что точность, с которой можно найти коэффициенты $w_C(y)$, оценивается сверху как $2^{O(n^2 \log n)} 2^{-\Omega(t)}$. При достаточно большом t , например, $t = n^3$, коэффициенты можно найти с точностью $1/2$. Поскольку коэффициенты весовой функции кода — целые числа, их можно найти точно.

Замечание Вычисление из предыдущего доказательства можно усилить. Пусть дан оракул, вычисляющий весовую функцию с точностью 2^{n-n^c} , $0 < c < 1$. Аналогично (11) можно написать

$$\left| \frac{w_C(y)(1+y)^t - \tilde{w}}{(1+y)^t} \right| = \left| w_C(y) - \frac{\tilde{w}}{(1+y)^t} \right| < 2^{(n+t)-(n+t)^c} |1+y|^{-t}.$$

В рассуждении из доказательства леммы 8 есть два параметра, которыми можно управлять: r и t . Повторяя это рассуждение, получаем следующее условие для этих параметров:

$$\left(2 \frac{n+1}{r} \right)^{n^2} n 2^{n^2} 2^{(n+t)-(n+t)^c} (2-r)^{-t} < \frac{1}{2}.$$

Можно проверить, что это условие выполняется при $t = \Omega(n^{3/c})$ и $r = 1/t$.

Для доказательства полноты приближенного вычисления весовой функции от фиксированного аргумента нам потребуется операция *сплетения* кодов. Сплетение $A \wr B$ кодов $A \subseteq \mathbb{F}_2^{n_1}$ и $B \subseteq \mathbb{F}_2^{n_2}$ — это подпространство тензорного произведения $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}$, которое является суммой подпространств $A \otimes 1^{n_2}$ и $1^{n_1} \otimes B$. Здесь 1^n обозначает вектор длины n , состоящий из одних единиц. В стандартном базисе векторы из $\mathbb{F}_2^{n_1} \otimes \mathbb{F}_2^{n_2}$ представляются матрицами размера $n_1 \times n_2$. В таком представлении код $A \wr B$ состоит из матриц c_{ab} , элементы которых имеют вид

$$c_{rs} = a_r \oplus b_s, \tag{12}$$

где $a = (a_1, \dots, a_{n_1}) \in A$, $b = (b_1, \dots, b_{n_2}) \in B$. Если хотя бы один из кодов A, B не содержит вектора из одних единиц, подпространства $A \otimes 1^{n_2}$ и $1^{n_1} \otimes B$ пересекаются только по нулевому вектору. В этом случае, который мы будем называть невырожденным, сплетение $[n_1, k_1]$ -кода и $[n_2, k_2]$ -кода есть $[n_1 n_2, k_1 + k_2]$ -код, в противном случае сплетение — $[n_1 n_2, k_1 + k_2 - 1]$ -код.

В невырожденном случае соответствие, задаваемое формулой (12), является взаимнооднозначным, а для весов кодовых слов a, b, c_{ab} имеем соотношение

$$|c_{ab}| = |a|(n_2 - |b|) + |b|(n_1 - |a|) = n_2|a| + |b|(n_1 - 2|a|).$$

Из него следует соотношение между весовыми функциями

$$w_{A \wr B}(y) = \sum_{a \in A} \sum_{b \in B} y^{n_2|a|} y^{|b|(n_1-2|a|)} = \sum_{a \in A} y^{n_2|a|} w_B(y^{n_1-2|a|}). \tag{13}$$

Доказательство леммы 9. Построим сведение задачи точного вычисления весовой функции w_C от $e^{\pi i/4p}$ к задаче приближенного вычисления от того же аргумента. Это удастся сделать при дополнительных ограничениях на константу α .

Без ограничения общности можно считать, что код C не содержит вектора из одних единиц. Проверка принадлежности вектора коду может быть выполнена за полиномиальное время. Если код содержит вектор из одних единиц, удлиним его, дописывая ко всем кодовым словам 0. Эта операция не меняет весовой функции кода.

Будем строить оценки высокой точности для значений весовой функции кода C от аргумента ω_p , используя оценки точности $2^{\alpha(n+a+t)}$ значений весовых функций кодов $C(a, t) = C_a \wr I_t$ от того же аргумента. Здесь I_t обозначает тривиальный код размерности (и длины) t , а кодовые слова кода C_a получаются дописыванием a нулей к каждому кодовому слову кода C . Заметим, что $w_{C_a} = w_C$. Из (13) получаем

$$w_{C(a,t)} = \sum_{x \in C} \omega_p^{t|x|} (1 + \omega_p^{n+a-2|x|})^t. \quad (14)$$

В дальнейшем считаем, что параметр t удовлетворяет условию $t \equiv 1 \pmod{8p}$, так что $\omega_p^t = \omega_p$.

Представим значение весовой функции кода C в виде суммы $4p$ слагаемых:

$$w_C(\omega_p) = \sum_{j=0}^{4p-1} M_j; \quad M_j = \sum_{x \in C, |x| \equiv j \pmod{4p}} \omega_p^{|x|}.$$

Очевидно, что $|M_j| < 2^n$.

Теперь перепишем (14) с учетом этого разложения на слагаемые:

$$w_{C(a,t)} = \sum_{j=0}^{4p-1} M_j (1 + e^{-\pi i j/2p} \omega_p^{n+a})^t. \quad (15)$$

Найдем значения M_j с большой точностью ε , решая систему линейных уравнений, которая получается следующим образом.

Выберем такое a , что $n + a = 2s + 1 \pmod{8p}$, $s = 0, \dots, 4p - 1$. Обозначим через \tilde{w}_s приближение с точностью $2^{\alpha(n+a+t)}$ для $w_{C(a,t)}(\omega_p)$ и $N = |1 + \omega_p|$. Из (15) получаем

$$\left| \sum_{j=0}^{4p-1} M_j \left(\frac{1 + e^{-\pi i j/2p} \omega_p^{2s+1}}{N} \right)^t - \frac{\tilde{w}_s}{N^t} \right| < 2^{\alpha(n+a+t)} / N^t = 2^{\alpha(n+a)} \left(\frac{2^\alpha}{N} \right)^t.$$

При $\alpha < \log_2 |1 + \omega_p|$ последний множитель мал и не превосходит $2^{-\Omega(t)}$. Заметим, что при $s \neq j, s \neq j-1 \pmod{4p}$ коэффициенты при M_j также малы:

$$\left| \left(\frac{1 + e^{-\pi i j / 2p} \omega_p^{2s+1}}{N} \right)^t \right| = \left(\frac{|1 + e^{\pi i (1+2(s-j))/4p}|}{N} \right)^t = 2^{-\Omega(t)}.$$

Поэтому при некотором достаточно большом t , которое тем не менее полиномиально ограничено n и $-\log \varepsilon$, мы получаем оценку μ_s (с точностью ε) величины $M_s + M_{s+1} \omega_p^{-1}$ ($s+1$ нужно брать по модулю $4p$).

Осталось учесть, что $p = O(1)$ (это параметр, характеризующий задачу). Решая (невыврожденную) систему уравнений

$$\begin{cases} x_0 + x_1 \omega_p^{-1} = \mu_0, \\ x_1 + x_2 \omega_p^{-1} = \mu_1, \\ \dots \\ x_{4p-1} + x_0 \omega_p^{-1} = \mu_{4p-1}, \end{cases}$$

получим $O(\varepsilon)$ -приближения для величин M_j . Знания этих величин достаточно для точного вычисления весовой функции кода C от аргумента ω_p .

ЛИТЕРАТУРА

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
2. Китаев А. Ю. Квантовые вычисления: алгоритмы и исправление ошибок // Успехи математических наук. 1997. Т. 52, № 6. С. 53–112.
3. Китаев А., Шень А., Вялый М. Классические и квантовые вычисления. М.: МЦНМО, 1999. (Английский перевод: **Kitaev A., Shen A., Vyalii M.** Classical and quantum computation. Providence, RI: AMS, 2002.)
4. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки. М.: Связь, 1979.
5. Лидл Р., Нидерейтер Х. Конечные поля. М.: Мир, 1989.
6. Boykin P. O., Mor T., Pulver M., Roychowdhury V. P., Vatan F. A new universal and fault-tolerant quantum basis // Inform. Process. Lett. 2000. V. 75, N 3. P. 101–107.
7. Dumer I., Micciancio D., Sudan M. Hardness of approximating the minimum distance of a linear code // Proc. of 40th annual symposium on foundations of computer science. NY: IEEE Computer Society, 1999. P. 475–485.

8. **Fortnow L., Rogers J.** Complexity limitations on quantum computation // J. Comput. and System Sci. 1999. V. 59, N 2. P. 240–252.
9. **Knill E., Laflamme R.** On the power of one bit of quantum computation. E-print, 1998. quant-ph/9802037.
10. **Nielsen M. A., Chuang I. L.** Quantum computation and quantum information. Cambridge: Cambridge University Press, 2000.
11. **Toda S.** PP is as hard as the polynomial-time hierarchy // SIAM J. Comput. 1991. V. 20, N 5. P. 865–877.
12. **Vardy A.** Algorithmic complexity in coding theory and the minimum distance problem // Proc. of 29h annual ACM symposium on the theory of computing. N.Y.: ACM, 1997. P. 92–109.

Адрес автора:
ВЦ им. А. А. Дородницына РАН,
Вавилова, 40,
119991 Москва, Россия.
E-mail: vyalyi@mccme.ru

Статья поступила
29 июня 2004 г.