Оптическое преобразование Фурье с прямым измерением фазы как метод вычислений

А.А. Демидов

Аннотация

В контексте вычислительных методов разбирается эксперимент по оптическому преобразованию Фурье с прямым измерением фазы (Nature 2017), производится корректная оценка его действительной вычислительной сложности с учётом помех, превосходящей возможности вычислительных методов, приводится методика эксперимента по прямому измерению вычислительной сложности.

1 Введение

В работе [1] был представлен метод оптического преобразования Фурье с прямым измерением фазы, позволяющий получить комплексные компоненты сигнала прямым измерением. Хотя заявленная авторами оценка вычислительной сложности $O(n)$ посчитана не строго, при аккуратном расчёте с учётом искажений оценка $\leqslant O(\log n)$ превышает возможности известных вычислительных методов $O(n \log n)$.

Что влечёт конфликт с физическим тезисом Чёрча-Тьюринга, утверждающим, что любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга. Это открывает возможности создания физического устройства, превосходящего машину Тьюринга (такого как квантовый компьютер).

Оценка $O(n \log n)$ вычислительного метода БПФ является эмпирической — не известно и не усматривается параллельного алгоритма, способного существенно её улучшить, хотя доказательства отсутствия такого алгоритма до сих пор не получено. В данной работе предпринимаются шаги для фиксации оценки $O(n \log n)$, в частности — предложен нетьюринговый табличный преобразователь Фурье (оракул).

Решение задачи факторизации с помощью квантовых компьютеров, по всей видимости, нарушает физический тезис Чёрча-Тьюринга. Отсутствие неограниченно масштабируемого квантового компьютера, экспоненциальная сложность создания которого пропорциональна необходимой мощности 2^q, не позволяет доказать это строго, поскольку любая *конечная* задача (даже алгоритмически неразрешимая проблема) разрешима конечным автоматом, тем более — машиной Тьюринга, таблично.

2 Базовые понятия теории алгоритмов

Понятие алгоритма, интуитивно определяемого как последовательность элементарных шагов, формально вводится посредством определения множества частично-рекурсивных функций вида $f\colon N \to N$, определённых на множестве натуральных чисел N. Частично-рекурсивными являются базовые функции: $f(x) = 0$; $f(x) = x + 1$; $f(i, x_0, \ldots, x_n) = x_i$ — и все, полученные из них с помощью примитивно-рекурсивных операторов *суперпозиции* $h(x_0, \ldots, x_n) = f\left(g_0(x_0, \ldots, x_n), \ldots, g_m(x_0, \ldots, x_n)\right)$ и *примитивной рекурсии* $h(i, x_0, \ldots, x_n) = f\left(h(i-1, x_0, \ldots, x_n)\right)$ при $h(0, x_0, \ldots, x_n) = g(x_0, \ldots, x_n)$, а также частично-рекурсивного оператора *минимизации аргумента*: $h(x_1, \ldots, x_n) = \min x_0 \big| f(x_0, \ldots, x_n) = 0$.

Классы частично-рекурсивных и вычислимых по Тьюрингу функций совпадают, поэтому алгоритм в теории вычислимости является частично-рекурсивной функцией. Оператор минимизации аргумента, соответствующий неограниченному циклу, определён не при всех значениях f и N, такие задачи являются алгоритмически неразрешимыми, то есть не имеющими решения *в общем случае*: для некоторых $n \in N$ решение не может быть найдено никогда. Поскольку множество алгоритмов счётно, множество задач, имеющих решение, имеет меру нуль на множестве всех задач — произвольных функций на N. Большинство задач из $2^N = \mathfrak{c}$ невозможно даже сформулировать в рамках одной грамматики с конечным алфавитом A и счётным множеством строк $|A^*| = \aleph_0$.

Вопросы интерпретации алгоритмов, описания задач в терминах естественного языка, выходят за рамки собственно теории алгоритмов, и изучаются в рамках формальных языков и грамматик, математической логики и теории моделей. Так, согласно теореме Лёвенгейма-Скулема, понятие мощности множества не является абсолютным, а зависит от модели: пока у нас нет биекции $N \mapsto 2^N$, два множества имеют разную мощность, но как только нам её привнесли — существование биекции даёт равномощность множеств по определению. Во взаимосвязи с теоремами Гёделя о неполноте эти вопросы чрезвычайно не просты.

Понятия неограниченного и бесконечного множества различны. Так, если запись натурального числа $n \in N$ имеет неограниченную, но конечную длину $\log n$ (всегда является числом), то верхней границы самого перенумерованного множества N не существует — оно бесконечно:

$$\lim_{n\to k}(n) = k, \quad \lim_{n\to\infty} \sum_{1}^{\log n} 1 = \ell \in N, \quad \lim_{n\to\infty}(n) = \aleph_0 \notin N, \quad \lim_{n\to\infty}(2^n) = \mathfrak{c}$$

Интуитивно это понимается так, что ограниченный цикл не превышает некоторого, наперёд заданного числа k. Неограниченный цикл всегда завершается, хотя число итераций ℓ заранее неизвестно. Бесконечный же цикл не завершается никогда — независимо от времени ожидания.

Замечание 2.1. *Распишем формулу для неограниченного множества:*

$$\lim_{n\to\infty}\sum_{1}^{\log n} 1 \sim \lim_{n\to\infty}\int_{1}^{\log n} dx\left[\log x = y\right] = \lim_{n\to\infty}\int_{1}^{n}\frac{1}{y}dy \sim \lim_{n\to\infty}\sum_{1}^{n}\frac{1}{n}$$

— это расходящийся гармонический ряд, однако предельный переход от суммы, где $n \in N$, к интегралу, где $x \in 2^N$, здесь не обоснован: в одном случае суммируются рациональные числа (конечные цепные дроби), а в другом — действительные (бесконечные цепные дроби). Известно, что «истончённый» гармонический ряд Кемпнера без некоторых членов сходится — если округлить члены до целых, ряд бы сошёлся к 2 *(этот парадокс интегрирования детально разобран в приложении А).*

Для каждой разрешимой задачи существует множество алгоритмов решения, среди которых существует оптимальный — решающий данную задачу за минимально возможное время, которое определяется числом шагов машины Тьюринга, эквивалентным количеству применений частично-рекурсивных функций и операторов. Зависимость времени решения задачи от размера входных данных $O(n)$ называют оценкой вычислительной сложности выбранного алгоритма. Оценка оптимального алгоритма определяет класс сложности задачи: P, NP и т.п.

Примитивно-рекурсивную функцию алгоритма невозможно задать таблично на всей области определения: это потребовало бы бесконечной ленты, которая лишь неограничена. Также существуют фундаментальные физические ограничения, связанные с конечностью скорости света: любое размещение данных в n ячейках требует 3D-пространства не менее $\sqrt[3]{n}$. Поэтому класс сложности задачи в тьюринговых вычислениях является понятием, определённым достаточно строго.

Параллельные алгоритмы сводятся к последовательным посредством многоленточных (фиксированное множество потоков) либо недетерминированных (неограниченное множество потоков) машин Тьюринга. В недетерминированной машине Тьюринга из текущего состояния s_i возможно множество S_i различных переходов, которые реализуются одновременно. В отличие от детерминированной машины Тьюринга, проходящей единственный «путь вычислений», недетерминированный вариант сразу проходит экспоненциальное число путей в «дереве вычислений». Любая задача, разрешимая на недетерминированной машине Тьюринга, разрешима и на детерминированной за счёт обхода дерева вычислений в ширину. Класс задач, выполняемых за полиномиальное время P на недетерминированной машине Тьюринга, называется NP.

При бесконечном множестве потоков недетерминированная модель не является машиной Тьюринга: при неограниченной высоте $h \to \ell$ дерева вычислений множество состояний 2^h бесконечно, бесконечна и лента,

которую машина Тьюринга никогда не сможет просмотреть целиком (в силу существования биекции узлов с бесконечным натуральным рядом).

Доказано [2], что время работы стандартной машины Тьюринга является не более чем квадратом времени p-ленточной $O(\mathfrak{f}_\mathfrak{p}(n)) \geqslant O(\sqrt{\mathfrak{f}_\mathfrak{1}(n)})$, и не более чем экспонентой, или даже лучше — это не доказано строго, времени недетермированной $O(\mathfrak{f}_\ell(n)) > O(\log \mathfrak{f}_\mathfrak{1}(n) - 1)$. Оценка зависит от способности алгоритма к распараллеливанию: $S(p,n) = \frac{\mathfrak{f}_\mathfrak{1}(n)}{\mathfrak{f}_\mathfrak{p}(n)}$ — проекция на ось n называется ускорением, а на p — масштабируемостью. Накладные расходы $\alpha \leqslant 1$ на синхронизацию и взаимодействие потоков

$$S(p) = \lim_{p\to\infty} \left(\alpha + \frac{1-\alpha}{p}\right)^{-1} = \alpha^{-1}$$

(закон Амдала) ограничивают оценку интервалом $O(\alpha\mathfrak{f}_\mathfrak{1}(n)) \dots O(\mathfrak{f}_\ell(n))$.

Машины Тьюринга с оракулом — нетьюринговым решателем, реализующим некоторую функцию из множества задач $2^N = \mathfrak{c}$ и вычисляющим её за один шаг $O(1)$ — позволяют, при подходящем выборе оракула, решать любые задачи, в том числе — алгоритмически неразрешимые.

Замечание 2.2. *Квантовая физика открытых систем в матричном представлении алгоритмически неразрешима, поскольку тензорное умножение (произведение Кронекера), задающее вектор состояния составной системы, увеличивает длину этого вектора как 2^n, поэтому при неограниченном росте системы его длина становится бесконечной.*

3 Табличный преобразователь (оракул)

Связь тьюринговых вычислений и измерений физических процессов устанавливает следующий табличный преобразователь с идеальной оценкой $O(1)$, позволяющий реализовать функцию любой задачи из 2^N, в том числе — алгоритмически невычислимой, на конечном интервале, на котором преобразователь реализует вычисления с оракулом.

Все оптические и подобные им вычисления, в основном, таковы, что процесс получения результата можно разделить на 2 фазы: подготовка оптического стенда и его освещение. Первая — вычислительно сложна, вторая — без учёта физических ограничений близка к оценке $O(1)$. Сконфигурированный стенд является оракулом некоторой задачи из 2^N.

Пусть входные данные представлены в виде массива A длины n, если необходимо — пополам для действительных и мнимых компонент. Каждый отсчёт a_i может принимать только значения $\{0, 1\}$.

Преобразователь состоит из массива F модулей ПЗУ (постоянной памяти) размера n — по одному модулю на каждый отсчёт входных данных, причём массив A является шиной адреса одновременно для всех модулей ПЗУ. Каждый модуль ПЗУ хранит таблицу значений и возвращает единственный отсчёт выходных данных, получая на адресный вход

весь массив входных данных. Электрический сигнал со входа A напрямую поступает на выход $F(A)$ после коммутации на ключах, записанных в ПЗУ. Одновременная коммутация за время $O(1)$ достигается при следующей организации адресации модуля ПЗУ (Рис. 1).

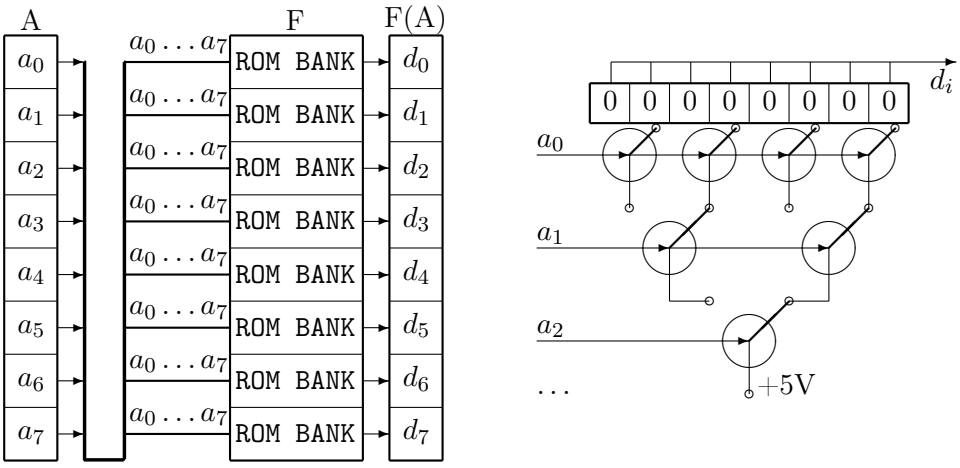

Рис. 1: Схема табличного преобразователя.

Преобразователь требует объёма массива ПЗУ $n \cdot 2^n$ относительно ширины шины адреса n (при неограниченной ширине — бесконечного объёма), поэтому величина n может быть только ограниченной.

С учётом «длины проводов», определяемой через объёмную плотность, реальная оценка преобразователя составляет $O(\sqrt[3]{2^n \cdot 2n})$ и мала в абсолютном выражении лишь для небольших n.

Действительно, объём модуля ПЗУ равен 2^n, всей сборки — $n \cdot 2^n$. Если каждый вентиль занимает объём μ^3, а их — по 2 на каждую ячейку памяти, то объём сборки $V = 2n \cdot 2^n \mu^3$. При наиболее плотной — объёмной упаковке проводников, размер устройства составит $\sqrt[3]{V}$, поэтому реальная оценка с учётом конечности скорости света c и времени C срабатывания одного ключа составляет $O(\mu \sqrt[3]{2^n \cdot 2n} + C)$. В силу малости μ: при $n = 64$ и техпроцессе $\mu = 1$ нм время $t = 4.4$ пс $+ C$ не превышает 1 такта вплоть до частоты процессора 100 ГГц.

4 Табличное преобразование Фурье

Сложность табличного преобразователя зависит от количества симметрий реализуемой функции: так, для реализации $f(x) = 0$ весь массив ПЗУ и вентилей можно исключить, оставив лишь вывод d_i (Рис. 1) — реальная оценка в этом случае равна идеальной $O(1)$. Симметрии преобразования Фурье так же позволяют упростить схему.

5 Физический смысл сложности задач

Оценки последовательного вычислительного метода преобразования Фурье, лучшей $O(n \log n)$ для БПФ, неизвестно [4]. Параллельные алгоритмы достигают $O\left(\frac{n}{p} \log n\right)$, где p — количество потоков [5, 6, 7]. Теоретическая оценка для недетерминированной машины Тьюринга лежит в интервале $O(n \log n) \ldots O(\log n + \log \log n - 1)$. Не доказано, что эта оценка является наилучшей — что сложность самой задачи составляет $O(\log n)$, тем более, не предложено алгоритмов с подобной эффективностью.

6 Прямое измерение фазы в эксперименте

Любую, в общем случае комплексную, функцию можно разложить на чётную (even) и нечётную (odd) составляющие по следующей формуле

$$f(x) = \underbrace{\frac{f(x) + f(-x)}{2}}_{even} + \underbrace{\frac{f(x) - f(-x)}{2}}_{odd}$$

Преобразование Фурье комплексной функции имеет симметрии относительно чётных и нечётных компонент [3]

Функция	**Образ**
Вещественная чётная	Вещественная чётная
Вещественная нечётная	Мнимая нечётная
Мнимая чётная	Мнимая чётная
Мнимая нечётная	Вещественная нечётная
Вещественная чётная + мнимая нечётная	Вещественная
Вещественная нечётная + мнимая чётная	Мнимая

A Гармонический ряд и интеграл

Список литературы

[1] Macfaden A.J., Gordon G.S.D., Wilkinson T.D. An optical Fourier transform coprocessor with direct phase determination // Sci Rep 7, 13667, 2017. DOI: 10.1038/s41598-017-13733-1.

[2] Хопкрофт Д.Э., Мотвани Р., Ульман Д.Д. Введение в теорию автоматов, языков и вычислений // М.: Издательский дом «Вильямс», 2-е изд. (пер. с англ.), 2008. — 528 с. ISBN: 978-5-907144-78-1.

[3] Bracewell R.N. The Fourier Transform and Its Applications // New York: McGraw-Hill, 1966. — 381 p. ISBN: 978-0073039381.

[4] Johnson S.G., Frigo M. A Modified Split-Radix FFT With Fewer Arithmetic Operations // IEEE Transactions on Signal Processing, vol. 55, No. 1, pp. 111–119, Jan. 2007. DOI: 10.1109/TSP.2006.882087.

[5] Thulasiraman P., Theobald K.B., Khokhar A.A., Gao G.R. Multithreaded algorithms for the fast Fourier transform // Proceedings of the 12th ACM symposium on Parallel algorithms and architectures, USA, NY: ACM, pp. 176–185, 2000. DOI: 10.1145/341800.341821.

[6] Gupta A., Kumar V. The scalability of FFT on parallel computers // IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 8, pp. 922–932, Aug. 1993. DOI: 10.1109/71.238626.

[7] Lui B. Parallel Fast Fourier Transform // Studies in Parallel and Distributed System. Auckland Campus, Massey University of New Zealand. 2009. — 9 p.