

УДК 519.1

Метод распознавания классов алгоритмов на основе асимптотики эластичности функций сложности

Валентина В.Быкова*

Институт математики,
Сибирский федеральный университет,
Свободный 79, Красноярск, 660041
Россия

Предложен новый признак выявления классов алгоритмов, основанный на асимптотическом поведении эластичности функций сложности. Использована существующая аналогия между функциями сложности алгоритмов и производственными функциями, темп роста которых в экономике традиционно оценивается эластичностью. Доказана теорема, устанавливающая характеристику эластичности для быстрых, полиномиальных, субэкспоненциальных, экспоненциальных и гиперэкспоненциальных алгоритмов. Основное достоинство предложенного признака — простота вычисления, обусловленная известными свойствами эластичности.

Ключевые слова: сложность вычислений, эластичность алгоритмов.

Введение

Стремительное повышение производительности современных компьютеров не снимает проблемы разработки алгоритмов, позволяющих за разумное время решать практически значимые NP -полные и NP -трудные задачи большой размерности [5]. Поэтому по-прежнему актуальна задача разработки методов классификации алгоритмов по вычислительной сложности. На специальные методы анализа и классификации алгоритмов опирается современная индустрия математического и программного обеспечения компьютерных систем, так как предполагает постоянный выбор эффективных алгоритмов и установление практической вычислимости решаемых задач.

Классификация задач по сложности — предмет исследования дескриптивной теории алгоритмов, а классификация алгоритмов — метрической. В рамках метрической теории алгоритмов анализ и систематизация алгоритмов осуществляются с точки зрения трудоемкости продуцируемых ими вычислительных процессов. При этом трудоемкость алгоритма формально описывается функцией временной сложности $t(n)$, отражающей максимальное количество элементарных шагов (в рамках принятой модели вычислений), которое необходимо алгоритму для достижения запланированного результата в зависимости от n — длины входа алгоритма или размерности решаемой задачи [7, 9]. Обычно ограничиваются рассмотрением поведения функций сложности в асимптотике при стремлении n к бесконечности, а изложение результатов, касающихся анализа и классификации алгоритмов, ведется в терминах O -большое и o -малое. Традиционное разделение алгоритмов на классы сложности — это выделение низкозатратных (полиномиальных) и высокозатратных (экспоненциальных) алгоритмов. Алгоритм, для которого $t(n) = O(n^k)$ или $t(n) = o(n^k)$, где k — неотрицательное действительное число, называют полиномиальным. Экспоненциальный алгоритм —

*e-mail: bykvalen@mail.ru

алгоритм, функция сложности которого растет быстрее любого алгебраического полинома. Инвариантность относительно модели вычислений — основная особенность данной классификации алгоритмов. Именно эта особенность позволяет подобным образом разделять задачи на полиномиально разрешимые и трудноразрешимые [9].

Современная алгоритмическая практика детализировала традиционную классификацию алгоритмов и выявила пять сложностных классов. Были выделены субполиномиальные алгоритмы из полиномиального класса, субэкспоненциальные и гиперэкспоненциальные алгоритмы из экспоненциального класса. В настоящее время к субполиномиальным (быстрым) принято относить алгоритмы, время выполнения которых при больших n растет медленнее любого полинома, а к субэкспоненциальным — алгоритмы, для которых порядок роста функций $t(n)$ выше любого полинома n^k , $k > 0$, и ниже всякой экспоненты $e^{\lambda n}$, $\lambda > 0$. К гиперэкспоненциальным причисляют алгоритмы с оценкой $e^{\lambda n} = o[t(n)]$, $\lambda > 0$. Быстрые и субэкспоненциальные алгоритмы — предмет повышенного внимания современных теоретических и прикладных исследований проблем информационной безопасности [2, 11]. Поэтому имеется потребность в простых и удобных для практики методах распознавания всех пяти сложностных классов алгоритмов. Использование непосредственного асимптотического оценивания для этих целей сопряжено в большинстве случаев с трудностями вычислительного характера. Известны попытки введения классификационных признаков, позволяющих распознавать современные классы алгоритмов. Например, авторы работы [3] предложили в качестве такого признака употреблять угловую меру асимптотического роста функций сложности, которая по определению числовая. Однако, как показали исследования [1], применение угловой меры в практике анализа алгоритмов ограничено в силу двух причин. Во-первых, вычисление данной меры порой затруднительно из-за громоздкости выкладок. Это имеет место даже в тех случаях, когда сложность исследуемых алгоритмов описывается простыми по виду логарифмически-экспоненциальными функциями (например, каскадами логарифмов, показательными полилогарифмами и т.п.). Во-вторых, угловая мера хотя и выделяет все используемые в данное время классы сложности алгоритмов, но сравнение алгоритмов, входящих в один и тот же класс, возможно не для всех сложностных классов. Так, все быстрые, субэкспоненциальные и гиперэкспоненциальные алгоритмы характеризуются постоянной угловой мерой (0 , $\pi/2$ и π соответственно). Сравнение таких алгоритмов между собой невозможно без непосредственного асимптотического оценивания. Таким образом, угловая мера асимптотического роста функций лишь частично решает проблему детальной классификации алгоритмов по вычислительной сложности, поэтому настоящая проблема остается пока открытой. В данной статье предлагается в качестве классификационного признака алгоритмов использовать асимптотическое поведение эластичности их функций сложности на бесконечности.

Соглашения относительно функций сложности

Формальный подход к классификации и анализу алгоритмов требует уточнения свойств рассматриваемых функций сложности $t(n)$. Относительно $t(n)$ далее действуют три соглашения. Во-первых, полагается, что $t(n)$ — монотонно неубывающие функции, областью значений которых выступает множество неотрицательных действительных чисел \mathbb{R}_+ , а областью определения — множество неотрицательных целых чисел \mathbb{Z}_+ . Это соглашение вполне соответствует природе $t(n)$ как функций, характеризующих трудоемкость выполнения алгоритмов. Во-вторых, допускается отступление от дискретности изменения n (с формальной

заменой n на x) — предположение, что аргумент $x \in \mathbb{R}_+$ и непрерывен, а необходимые значения функции $t(n)$ вычисляются в целочисленных точках $x = n$. Данное соглашение продиктовано необходимостью привлечения к исследованию алгоритмов инструментов математического анализа. В-третьих, рассматриваемое множество функций сложности ограничивается семейством “по существу положительных” логарифмически-экспоненциальных \mathcal{L} -функций.

Напомним, что семейство \mathcal{L} — это множество функций, введенное Г.Х. Харди [10]. Семейство \mathcal{L} определяется рекурсивно по правилам: постоянная $f(x) = c$ лежит в \mathcal{L} для всех действительных c ; тождественная функция $f(x) = x$ входит в \mathcal{L} ; разность $f(x) - q(x)$ двух \mathcal{L} -функций также является \mathcal{L} -функцией; если $f(x) \in \mathcal{L}$, то $e^{f(x)} \in \mathcal{L}$; если $f(x) \in \mathcal{L}$ и является “по существу положительной” функцией, то $\ln f(x)$ относится к \mathcal{L} . Функция $f(x)$ считается “по существу положительной”, если существует x_0 такое, что $f(x) > 0$ для всех $x > x_0$. Из данных правил следует, что если $f(x), q(x) \in \mathcal{L}$, то

$$f(x) + q(x) = f(x) - (0 - q(x)) \in \mathcal{L};$$

если $f(x)$ и $q(x)$ — “по существу положительные” \mathcal{L} -функции, то

$$f(x) \cdot q(x) = e^{\ln f(x) + \ln q(x)} \in \mathcal{L}, \quad f(x)/q(x) = e^{\ln f(x) - \ln q(x)} \in \mathcal{L};$$

любая степенная функция $x^k = e^{k \ln x} \in \mathcal{L}$ при $x > 1, k > 0$; для “по существу положительных” \mathcal{L} -функций $u = u(x)$ и $w = w(x)$ показательно-степенная функция общего вида u^w входит в семейство \mathcal{L} .

Известно, что каждая \mathcal{L} -функция непрерывна и дифференцируема в той области, где она определена. Кроме того, всякая \mathcal{L} -функция является либо “по существу положительной”, либо “по существу отрицательной”, либо тождественно равной нулю. Главная теорема Харди о \mathcal{L} -функциях констатирует, что эти функции образуют асимптотическую иерархию [4, 10]: *если $f(x), q(x) \in \mathcal{L}$, то при $x \rightarrow \infty$ либо $f(x) \prec q(x)$, либо $f(x) \succ q(x)$, либо $f(x) = O[q(x)]$* . Заметим, что используемая здесь и далее O -символика трактуется в смысле асимптотической пропорциональности функций:

$$f(x) = O[q(x)] \Leftrightarrow f(x) \sim cq(x) \Leftrightarrow \lim_{x \rightarrow \infty} \frac{f(x)}{q(x)} = c > 0.$$

Кроме того, для сравнения двух функций по скорости роста на бесконечности далее применяются отношение Поль-Дюбуа Раймона \prec и o -малое:

$$f(x) \prec q(x) \Leftrightarrow f(x) = o[q(x)] \Leftrightarrow \lim_{x \rightarrow \infty} \frac{f(x)}{q(x)} = 0.$$

Установление границ рассматриваемого множества функций в виде семейства монотонно неубывающих, “по существу положительных” \mathcal{L} -функций обеспечивает существование эластичности и возможность сравнения любых двух функций сложности алгоритмов по скорости роста. Важно, что всякая такая функция — либо отличная от нуля постоянная (тождественная или асимптотическая), либо положительная бесконечно большая величина при $x \rightarrow \infty$. Далее везде под \mathcal{L} -функциями понимаются именно такие функции. Введенные соглашения не ограничивают практическую значимость представленных далее результатов, поскольку функции сложности почти всех реальных алгоритмов входят в \mathcal{L} .

Эластичность и ее свойства

Производственной функцией в эконометрике называют аналитическое соотношение, связывающее переменные величины затрат (ресурсов) с объемом выпуска продукции. Для оценки поведения производственных функций обычно применяют различные дифференциальные характеристики. Наиболее популярная из них — эластичность. Понятие эластичности было введено А.Маршаллом [6] для анализа функции спроса. По существу, это понятие является чисто математическим и может быть применено для исследования любых непрерывных и дифференцируемых функций. Предлагаемая в настоящей работе область приложения эластичности — выделение сложностных классов алгоритмов. В самом деле, функция сложности алгоритма по своему назначению схожа с производственной функцией. Она указывает, как зависит количество необходимых для выполнения алгоритма вычислительных ресурсов от размерности задачи (объема или длины исходных данных). В этом смысле функция сложности описывает обратную зависимость, нежели производственная функция: если производственная функция — это связь “ресурсы → объем”, то функция сложности алгоритма — “объем → ресурсы”.

Под эластичностью $E_x(y)$ функции $y = f(x)$ принято понимать предел отношения относительного приращения этой функции к относительному приращению аргумента [8]:

$$E_x(y) = \lim_{\Delta x \rightarrow 0} \left(\frac{\Delta y}{y} : \frac{\Delta x}{x} \right) = \frac{x}{y} \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{x}{y} y' = x (\ln y)' = \frac{(\ln y)'}{(\ln x)'}. \quad (1)$$

Ясно, что для дифференцируемой, монотонно неубывающей функции $y = f(x) > 0$ на $(0, \infty)$ всегда существует эластичность и $E_x(y) \geq 0$. Согласно (1) при достаточно малых Δx справедливо приближение

$$\frac{\Delta y}{y} \approx E_x(y) \frac{\Delta x}{x},$$

которое означает, что эластичность $E_x(y)$ — коэффициент пропорциональности между темпами роста величин y и x .

Непосредственное применение (1) дает формулы эластичностей основных элементарных функций:

$$E_x(\ln x) = \frac{1}{\ln x}, \quad E_x(x^k) = k, \quad k > 0, \quad E_x(e^{\lambda x}) = \lambda x, \quad \lambda > 0.$$

Справедливы свойства, значительно упрощающие процесс вычисления эластичностей [8].

1. Всякая постоянная $y = c > 0$ имеет нулевую эластичность.
2. Эластичность — безразмерная величина:

$$E_x(y) = E_{ax}(by).$$

3. Эластичность обратной функции $x = f^{-1}(y)$ — обратная величина:

$$E_y(x) = \frac{1}{E_x(y)}.$$

4. Эластичность произведения функций $u = u(x)$ и $w = w(x)$ равна сумме их эластичностей:

$$E_x(u \cdot w) = E_x(u) + E_x(w).$$

5. Эластичность отношения функций $u = u(x)$ и $w = w(x)$ равна разности их эластичностей:

$$E_x(u/w) = E_x(u) - E_x(w).$$

6. Эластичность суммы функций $u = u(x)$ и $w = w(x)$ — сумма эластичностей слагаемых, взятых с соответствующими весами:

$$E_x(u+w) = \frac{u}{(u+w)} E_x(u) + \frac{w}{(u+w)} E_x(w).$$

7. Эластичность показательно-степенной функции общего вида $y = u^w$, где $u = u(x)$ и $w = w(x)$, задается соотношением

$$E_x(y) = w(E_x(w) \ln u + E_x(u)).$$

В частности,

$$E_x[e^{w(x)}] = w E_x(w). \quad (2)$$

8. Эластичность композиции функций $y = f(w)$ и $w = w(x)$ равна

$$E_x(y) = E_w(f) \cdot E_x(w).$$

Данные свойства приводят к следующим формулам эластичностей каскадов логарифмов, “башен” и экспонент общего вида соответственно:

$$E_x(\underbrace{\ln \ln \dots \ln x}_{k \text{ раз}}) = \frac{1}{\ln x \cdot \ln \ln x \cdot \dots \cdot \underbrace{\ln \ln \dots \ln x}_{k \text{ раз}}}, \quad k \geq 1;$$

$$E_x(\underbrace{2^{2^{\cdot^{\cdot^{2^x}}}}}_{k \text{ этажей}}) = x 2^x 2^{2^x} \dots \underbrace{2^{2^{\cdot^{\cdot^{2^x}}}}}_{k-1 \text{ этажей}} (\ln 2)^k, \quad k \geq 1;$$

$$E_x(e^{\lambda x^{w(x)}}) = \lambda x^{w(x)} w(x) (E_x(w) \ln x + 1).$$

Для функций субэкспоненциального порядка роста

$$y = e^{\lambda x^\xi (\ln x)^{1-\xi}}, \quad \lambda > 0, \quad 0 < \xi < 1, \quad (3)$$

описывающих сложность многих распространенных теоретико-числовых алгоритмов [2, 11], эластичность равна

$$E_x(e^{\lambda x^\xi (\ln x)^{1-\xi}}) = \lambda x^\xi (\ln x)^{1-\xi} \left(\xi + \frac{1-\xi}{\ln x} \right) = O[x^\xi (\ln x)^{1-\xi}] = o(x).$$

Утверждение 1. Эластичности \mathcal{L} -функций — тождественно постоянные, бесконечно малые или бесконечно большие при $x \rightarrow \infty$ логарифмически-экспоненциальные функции, образующие асимптотическую иерархию. Для монотонно неубывающей \mathcal{L} -функции всегда

$$E_x(y) \geq 0, \quad E_x(y) = o\left(\frac{y}{\ln x}\right).$$

Асимптотическое поведение эластичности \mathcal{L} -функции $y = f(x)$ инвариантно относительно полиномиального преобразования, заключающегося в умножении функции на константу $c > 0$ и возведению в положительную и отделенную от нуля степень $k > 0$, т.е. $E_x(cy^k) = O[E_x(y)]$.

Верность утверждения 1 вытекает из определения и свойств эластичности, правил построения семейства \mathcal{L} -функций и упомянутой главной теоремы Харди. \square

Утверждение 2. Пусть $1 \prec f(x)$ и $1 \prec q(x)$, т.е. $\lim_{x \rightarrow \infty} f(x) = \infty$, $\lim_{x \rightarrow \infty} q(x) = \infty$. Тогда если $\ln f(x) \prec \ln q(x)$, то $f(x) \prec q(x)$.

Это известное в математическом анализе высказывание, называемое леммой о логарифмическом пределе [12].

Утверждение 3. Пусть $f(x), q(x) \in \mathcal{L}$ и $1 \prec f(x)$, $1 \prec q(x)$. Тогда если $E_x(f) \prec E_x(q)$, то $f(x) \prec q(x)$.

Доказательство. Действительно, по условию данного утверждения,

$$\lim_{x \rightarrow \infty} \frac{E_x(f)}{E_x(q)} = \lim_{x \rightarrow \infty} \frac{x (\ln f(x))'}{x (\ln q(x))'} = \lim_{x \rightarrow \infty} \frac{(\ln f(x))'}{(\ln q(x))'} = 0.$$

Кроме того, $f(x)$ и $q(x)$ — бесконечно большие величины. Значит, $\ln f(x)$ и $\ln q(x)$ — также бесконечно большие при $x \rightarrow \infty$. Поэтому

$$\lim_{x \rightarrow \infty} \frac{(\ln f(x))'}{(\ln q(x))'} = 0 \quad \Rightarrow \quad \lim_{x \rightarrow \infty} \frac{\ln f(x)}{\ln q(x)} = 0.$$

Отсюда $\ln f(x) \prec \ln q(x)$, и по лемме о логарифмическом пределе, $f(x) \prec q(x)$. \square

Утверждение 3 свидетельствует о том, что иерархия эластичностей порождает идентичную иерархию \mathcal{L} -функций. Между тем обратное не всегда верно. Как будет показано далее, если $f(x) \prec q(x)$, то $E_x(f) \prec E_x(q)$ или $E_x(f) = O[E_x(q)]$.

Шесть классов \mathcal{L} -функций

Очевидно, что если функция $f(x) \in \mathcal{L}$, то она представима в виде экспоненты $f(x) = e^{w(x)}$, где $w(x)$, например, задается выражением $w(x) = \ln f(x) \in \mathcal{L}$. Такие экспоненты образуют асимптотическую иерархию, при этом

$$e^{w_1(x)} \prec e^{w_2(x)} \quad \Leftrightarrow \quad \lim_{x \rightarrow \infty} [w_1(x) - w_2(x)] = -\infty.$$

Отсюда вытекает справедливость следования

$$1 \prec w_1(x) \prec w_2(x) \quad \Rightarrow \quad e^{w_1(x)} \prec e^{w_2(x)}.$$

Руководствуясь данным положением и асимптотическим подходом, определим на рассматриваемом семействе \mathcal{L} шесть классов функций, с принципиально различным порядком роста:

- асимптотические константы

$$\mathbf{Const} = \{f(x) \mid f(x) = O(1)\};$$

- функции субполиномиального порядка роста

$$\mathbf{Subpoly} = \left\{f(x) \mid 1 \prec f(x) \prec e^{O(\ln x)}\right\};$$

- функции полиномиального порядка роста

$$\mathbf{Poly} = \left\{ f(x) \mid f(x) = O\left[e^{O(\ln x)}\right] \right\};$$

- функции субэкспоненциального порядка роста

$$\mathbf{Subexp} = \left\{ f(x) \mid e^{O(\ln x)} \prec f(x) \prec e^{O(x)} \right\};$$

- функции экспоненциального порядка роста

$$\mathbf{Exp} = \left\{ f(x) \mid f(x) = O\left[e^{O(x)}\right] \right\};$$

- функции гиперэкспоненциального порядка роста

$$\mathbf{Hyperexp} = \left\{ f(x) \mid e^{O(x)} \prec f(x) \right\}.$$

Класс **Const** составляют функции, значения которых на бесконечности ограничены сверху некоторой отличной от нуля константой. Это, прежде всего, асимптотические константы. Конечно, к классу **Const** относятся все тождественно постоянные \mathcal{L} -функции.

Классу **Subpoly** принадлежат функции, которые не входят в класс **Const** и которые растут медленнее любого полинома. Например, это каскады логарифмов, показательные полилогарифмы

$$x^{\lambda(\ln x)^{m-1}} = e^{\lambda(\ln x)^m}, \quad \lambda > 0, \quad 0 < m < 1,$$

полилогарифмы

$$(\ln x)^m = e^{m \ln \ln x}, \quad m > 0.$$

Класс **Poly** преимущественно состоит из обычных полиномов

$$f(x) = O(x^k) = O(e^{k \ln x}), \quad k > 0.$$

Он также включает в себя обобщенные полиномы

$$f(x) = O(e^{k \ln x} q(x)), \quad q(x) \in \mathbf{Const}, \mathbf{Subpoly}.$$

Класс **Poly** допускает определение не только через O -большое, но и с помощью отношения Поль-Дюбуа Раймона:

$$\mathbf{Poly} = \left\{ f(x) \mid \exists k > 0 : e^{(k-\varepsilon) \ln x} \prec f(x) \prec e^{(k+\varepsilon) \ln x}, \quad \varepsilon \rightarrow +0 \right\}. \quad (4)$$

Класс **Subexp** образуют функции, которые растут быстрее любого полинома и медленнее всякой показательной функции. В класс **Subexp**, в частности, входят функции вида (3), показательные полилогарифмы

$$(\ln x)^{m \ln x} = e^{m \ln x \ln \ln x}, \quad m > 0,$$

показательные полилогарифмы

$$x^{\lambda(\ln x)^{m-1}} = e^{\lambda(\ln x)^m}, \quad \lambda > 0, \quad m > 1.$$

Класс **Exp** содержит показательную функцию $y = O(e^{\lambda x})$, $\lambda > 0$, а также обобщенные экспоненты

$$f(x) = O(e^{\lambda x}) q(x), \quad q(x) \in \mathbf{Const}, \mathbf{Subpoly}, \mathbf{Poly}, \mathbf{Subexp}.$$

Этот класс функций можно задать с помощью отношения Поль-Дюбуа Раймона следующим образом:

$$\mathbf{Exp} = \left\{ f(x) \mid \exists \lambda > 0 : e^{\lambda x^{1-\varepsilon}} \prec f(x) \prec e^{\lambda x^{1+\varepsilon}}, \varepsilon \rightarrow +0 \right\}. \quad (5)$$

Так, классу **Exp** принадлежат обобщенная экспонента $y = e^{x+(\ln x)^2}$, одноэтажная “башня” $2^x = e^{x \ln 2}$. Показательно-степенная функция $x^x = e^{x \ln x}$ и “башни” с двумя и более этажами входят уже в класс **Hyperexp**. Функции этого класса растут быстрее, чем всякая экспонента вида $e^{O(x)}$.

Леммы об эластичности различных классов \mathcal{L} -функций

Лемма 1. Если $f(x) \in \mathbf{Const}$, то $E_x(f) = 0$ или $E_x(f) = o(1)$.

Доказательство. Если $y = f(x)$ – тождественная постоянная, то по свойству 1 эластичности (эластичность константы) $E_x(f) = 0$. Пусть теперь $y = O(1)$. Поскольку $\ln x E_x(y) = o(y)$ в силу утверждения 1, то при $x \rightarrow \infty$

$$E_x [O(1)] = \frac{1}{\ln x} o(1) = o(1). \quad (6)$$

□

Лемма 2. Функция $f(x) \in \mathbf{Poly} \Leftrightarrow E_x(f) = O(1)$.

Доказательство. Пусть $f(x) = e^{O(\ln x)}$. Учитывая свойство 4 эластичности (эластичность произведения), формулу (2) и оценку (6), имеем:

$$E_x [O(\ln x)] = E_x [(\ln x) O(1)] = E_x(\ln x) + E_x [O(1)] = \frac{1}{\ln x} + \frac{1}{\ln x} o(1); \quad (7)$$

$$\begin{aligned} \lim_{x \rightarrow \infty} E_x \left[e^{O(\ln x)} \right] &= \lim_{x \rightarrow \infty} O(\ln x) E_x [O(\ln x)] = \\ &= \lim_{x \rightarrow \infty} (\ln x) O(1) \left(\frac{1}{\ln x} + \frac{1}{\ln x} o(1) \right) = O(1). \end{aligned} \quad (8)$$

Пусть $f(x) = O[e^{O(\ln x)}] = e^{O(\ln x)} O(1)$ при $x \rightarrow \infty$. Тогда

$$\lim_{x \rightarrow \infty} E_x(f) = \lim_{x \rightarrow \infty} E_x \left[e^{O(\ln x)} O(1) \right] = \lim_{x \rightarrow \infty} E_x \left[e^{O(\ln x)} \right] + \lim_{x \rightarrow \infty} E_x [O(1)] = O(1).$$

Докажем справедливость обратного утверждения. Пусть $f(x)$ – функция с асимптотически постоянной эластичностью, т.е.

$$\lim_{x \rightarrow \infty} E_x(f) = c > 0.$$

Воспользовавшись определением эластичности и правилом Лопиталья, получаем

$$\lim_{x \rightarrow \infty} E_x(f) = \lim_{x \rightarrow \infty} \frac{(\ln f(x))'}{(\ln x)'} = \lim_{x \rightarrow \infty} \frac{\ln f(x)}{\ln x} = c > 0. \quad (9)$$

Существование конечного предела (9) означает: для произвольного сколь угодно малого $\varepsilon > 0$ найдется такое $\delta(\varepsilon) > 0$, что при $x > \delta(\varepsilon)$ будут выполняться неравенства

$$c - \varepsilon < \frac{\ln f(x)}{\ln x} < c + \varepsilon.$$

С учетом монотонности логарифма и экспоненты данные неравенства преобразуются к виду

$$e^{(c-\varepsilon)\ln x} < f(x) < e^{(c+\varepsilon)\ln x}.$$

В этих условиях всегда можно подобрать константу $k > 0$ и ничтожно малую постоянную $\varepsilon > 0$ так, чтобы

$$e^{(k-\varepsilon)\ln x} \prec f(x) \prec e^{(k+\varepsilon)\ln x}.$$

Но такие функции, согласно (4), находятся в классе **Poly**. □

Лемма 3. Если функция $f(x) \in \mathbf{Subpoly}$, то $E_x(f) = o(1)$. Если $E_x(f) = o(1)$, то $f(x) \in \mathbf{Subpoly}$ или $f(x) \in \mathbf{Const}$.

Доказательство. Пусть $f(x) \in \mathbf{Subpoly}$, т. е. $1 \prec f(x) \prec e^{O(\ln x)}$. Это значит, что $f(x)$ — бесконечно большая величина, и ее можно представить в виде

$$f(x) = O\left[e^{O(\ln x)/\gamma(x)}\right] = e^{O(\ln x)/\gamma(x)}O(1), \quad 1 \prec \gamma(x), \quad \gamma(x) = o(\ln x), \quad (10)$$

где $\gamma(x) \in \mathcal{L}$. Отметим, что в представлении (10) условие $\gamma(x) = o(\ln x)$ необходимо, чтобы исключить случай $f(x) = O(1)$, так как $1 \prec f(x)$. По утверждению 1 для $\gamma(x) \in \mathcal{L}$ верна оценка

$$\frac{(\ln x) E_x(\gamma)}{\gamma(x)} = o(1). \quad (11)$$

Применение к (10) свойств эластичности, оценок (6), (7), (11) приводит к пределу

$$\begin{aligned} \lim_{x \rightarrow \infty} E_x(f) &= \lim_{x \rightarrow \infty} E_x\left[e^{O(\ln x)/\gamma(x)}O(1)\right] = \lim_{x \rightarrow \infty} E_x\left[e^{O(\ln x)/\gamma(x)}\right] + \lim_{x \rightarrow \infty} E_x[O(1)] = \\ &= \lim_{x \rightarrow \infty} E_x\left[e^{O(\ln x)/\gamma(x)}\right] + o(1) = \lim_{x \rightarrow \infty} O(\ln x)/\gamma(x) E_x[O(\ln x)/\gamma(x)] = \\ &= \lim_{x \rightarrow \infty} \frac{\ln x O(1)}{\gamma(x)} (E_x[O(\ln x)] - E_x(\gamma)) = \\ &= \lim_{x \rightarrow \infty} \frac{\ln x}{\gamma(x)} \left(\frac{1}{\ln x} + \frac{1}{\ln x} o(1) - E_x(\gamma) \right) = \\ &= \lim_{x \rightarrow \infty} \frac{1}{\gamma(x)} (1 + o(1)) - \lim_{x \rightarrow \infty} \frac{\ln x E_x(\gamma)}{\gamma(x)} = 0. \end{aligned}$$

Следовательно, $E_x(f) = o(1)$.

Пусть $E_x(f) = o(1)$. Учитывая (8), и подавно $E_x(f) = o[E_x(e^{O(\ln x)})]$, т. е. $E_x(f) \prec E_x(e^{O(\ln x)})$. Отсюда по утверждению 3 получаем $f(x) \prec e^{O(\ln x)}$. Такие функции принадлежат **Const** или **Subpoly**. □

Лемма 4. Функция $f(x) \in \mathbf{Exp} \Leftrightarrow E_x(f) = O(x)$.

Доказательство. Предположим, что $f(x) = e^{O(x)}$. Выполняя рассуждения, аналогичные рассуждениям леммы 2, и используя оценку (6), имеем

$$E_x[O(x)] = E_x[xO(1)] = E_x(x) + E_x[O(1)] = 1 + \frac{1}{\ln x} o(1); \quad (12)$$

$$\lim_{x \rightarrow \infty} E_x[e^{O(x)}] = \lim_{x \rightarrow \infty} O(x) E_x[O(x)] = \lim_{x \rightarrow \infty} O(x) \left(1 + \frac{1}{\ln x} o(1)\right) = O(x). \quad (13)$$

Пусть теперь $f(x) = O[e^{O(x)}] = e^{O(x)}O(1)$. Применение (6), (13) дает

$$\lim_{x \rightarrow \infty} E_x(f) = \lim_{x \rightarrow \infty} E_x[e^{O(x)}O(1)] = \lim_{x \rightarrow \infty} E_x[e^{O(x)}] + \lim_{x \rightarrow \infty} E_x[O(1)] = O(x).$$

Докажем обратное. Допустим, что $E_x(f) = O(x)$, т.е. $E_x(f) = O(e^{\ln x}) \in \mathbf{Poly}$. Тогда в соответствии с (4) всегда можно подобрать ничтожно малое $\varepsilon > 0$ так, чтобы

$$e^{(1-\varepsilon)\ln x} \prec E_x(f) \prec e^{(1+\varepsilon)\ln x}$$

или, то же самое,

$$x^{1-\varepsilon} \prec E_x(f) \prec x^{1+\varepsilon}.$$

Если $E_x(f) \prec x^{1+\varepsilon}$, то по утверждению 3

$$E_x(f) = o(x^{1+\varepsilon}) = o\left[\frac{1}{\lambda(1+\varepsilon)} E_x(e^{\lambda x^{1+\varepsilon}})\right] = o\left[E_x(e^{\lambda x^{1+\varepsilon}})\right] \Rightarrow f(x) \prec e^{\lambda x^{1+\varepsilon}}.$$

Аналогично, если $x^{1-\varepsilon} \prec E_x(f)$, то

$$x^{1-\varepsilon} = \frac{1}{\lambda(1-\varepsilon)} E_x(e^{\lambda x^{1-\varepsilon}}) = o[E_x(f)] \Rightarrow e^{\lambda x^{1-\varepsilon}} \prec f(x).$$

Это отвечает (5) и $f(x) \in \mathbf{Exp}$. □

Лемма 5. Функция $f(x) \in \mathbf{Subexp} \Leftrightarrow 1 \prec E_x(f) \prec x$.

Доказательство. Пусть $f(x) \in \mathbf{Subexp}$. Поскольку $f(x) \prec e^{O(x)}$, то функция $f(x)$ представима в виде

$$f(x) = O\left[e^{O(x)/\gamma(x)}\right] = e^{O(x)/\gamma(x)}O(1), \quad 1 \prec \gamma(x), \quad \gamma(x) = o(x), \quad (14)$$

где $\gamma(x) \in \mathcal{L}$. Ограничение $\gamma(x) = o(x)$ введено, чтобы исключить случай, когда $f(x)$ становится убывающей функцией. Из (14), свойств эластичности и оценок (6), (11), (12) имеем

$$\begin{aligned} E_x(f) &= E_x\left[e^{O(x)/\gamma(x)}O(1)\right] = E_x\left[e^{O(x)/\gamma(x)}\right] + E_x[O(1)] = \\ &= E_x\left[e^{O(x)/\gamma(x)}\right] + o(1) = O(x)/\gamma(x) E_x[O(x)/\gamma(x)] + o(1) = \\ &= \frac{xO(1)}{\gamma(x)} \left(1 + \frac{1}{\ln x} o(1) - E_x(\gamma)\right) + o(1), \end{aligned}$$

$$\lim_{x \rightarrow \infty} \frac{E_x(f)}{x} = \lim_{x \rightarrow \infty} \frac{O(1)}{\gamma(x)} \left(1 + \frac{1}{\ln x} o(1) - E_x(\gamma)\right) + o(1) = \lim_{x \rightarrow \infty} \frac{o(1)}{\ln x} = 0.$$

Это означает, что $E_x(f) \prec x$.

Предположим, что $e^{O(\ln x)} \prec f(x)$. Отсюда

$$f(x) = O\left[e^{O(\ln x)\gamma(x)}\right] = e^{O(\ln x)\gamma(x)}O(1), \quad 1 \prec \gamma(x),$$

где $\gamma(x) \in \mathcal{L}$. Тогда из неравенства $E_x(\gamma) \geq 0$, свойств эластичности и оценок (6), (7) следует, что

$$\begin{aligned} E_x(f) &= E_x \left[e^{O(\ln x)\gamma(x)} O(1) \right] = E_x \left[e^{O(\ln x)\gamma(x)} \right] + E_x [O(1)] = \\ &= O(\ln x) \gamma(x) E_x [O(\ln x) \gamma(x)] + o(1) = \\ &= \ln x O(1) \gamma(x) \left(\frac{1}{\ln x} + \frac{1}{\ln x} o(1) + E_x(\gamma) \right) + o(1) = \\ &= O(1) \gamma(x) (1 + o(1) + \ln x E_x(\gamma)) + o(1), \\ \lim_{x \rightarrow \infty} \frac{1}{E_x(f)} &= \lim_{x \rightarrow \infty} \frac{1}{O(1) \gamma(x) (1 + o(1) + \ln x E_x(\gamma)) + o(1)} = \\ &= \lim_{x \rightarrow \infty} \frac{1}{\gamma(x) (1 + \ln x E_x(\gamma))} = 0. \end{aligned}$$

Справедливость обратного положения вытекает из утверждения 3. В самом деле, если $E_x(f) = o(x)$, то согласно (13) $E_x(f) = o[E_x(e^{O(x)})]$. Следовательно, $f(x) \prec e^{O(x)}$. Если $1 = o[E_x(f)]$, то с учетом (8) также $E_x(e^{O(\ln x)}) = o[E_x(f)]$. Таким образом, $e^{O(\ln x)} \prec f(x)$. \square

Лемма 6. Функция $f(x) \in \mathbf{Hyperexp} \Leftrightarrow x \prec E_x(f)$.

Доказательство этой леммы подобно доказательству леммы 5 и основано на представлении функции $f(x) \in \mathbf{Hyperexp}$ в виде $f(x) = O[e^{O(x)\gamma(x)}]$, где $\gamma(x) \in \mathcal{L}$ и $1 \prec \gamma(x)$.

Теорема о классификации \mathcal{L} -функций

Из лемм 1–6 вытекает важный вывод: различным классам \mathcal{L} -функций свойственно принципиально различное поведение эластичности и, наоборот, по асимптотике поведения эластичности \mathcal{L} -функции можно определить класс, к которому относится эта функция. Данное соответствие однозначно, если объединить **Const**, **Subpoly** в один класс **Subpoly***. Этот класс будет содержать все \mathcal{L} -функции, которые, если и растут с увеличением аргумента, то растут медленнее любого полинома. Среди них возможны как асимптотически постоянные, так и бесконечно большие величины. Теперь результаты лемм 1–6 и утверждения 1 можно объединить в следующую теорему.

Теорема 1. Разбиение семейства монотонно неубывающих, “по существу положительных” \mathcal{L} -функций на классы **Subpoly***, **Poly**, **Subexp**, **Exp**, **Hyperexp** в соответствии с порядком их роста эквивалентно надлежащему разбиению по асимптотике эластичности этих функций на бесконечности:

$$\begin{aligned} \mathbf{Subpoly}^* &= \{f(x) \mid f(x) \prec e^{O(\ln x)}\} \equiv \{f(x) \mid E_x(f) = o(1)\}; \\ \mathbf{Poly} &= \{f(x) \mid f(x) = O[e^{O(\ln x)}]\} \equiv \{f(x) \mid E_x(f) = O(1)\}; \\ \mathbf{Subexp} &= \{f(x) \mid e^{O(\ln x)} \prec f(x) \prec e^{O(x)}\} \equiv \{f(x) \mid 1 \prec E_x(f) \prec x\}; \\ \mathbf{Exp} &= \{f(x) \mid f(x) = O[e^{O(x)}]\} \equiv \{f(x) \mid E_x(f) = O(x)\}; \\ \mathbf{Hyperexp} &= \{f(x) \mid e^{O(x)} \prec f(x)\} \equiv \{f(x) \mid x \prec E_x(f)\}. \end{aligned}$$

Данное разбиение инвариантно относительно полиномиального преобразования, заключающегося в умножении \mathcal{L} -функции на константу $c > 0$ и возведению в положительную и отделенную от нуля степень $k > 0$.

Классификация алгоритмов по асимптотике эластичности функций сложности

Теорема о классификации \mathcal{L} -функций позволяет систематизировать алгоритмы по асимптотическому поведению эластичности функций сложности. Пусть $t(n)$ — функция сложности алгоритма, где n — размерность задачи. Для выяснения принадлежности алгоритма к классу сложности требуется формальный переход от дискретного аргумента n к непрерывному x . Применительно к функции $t(x)$ эластичность $E_x(t)$ — коэффициент пропорциональности между темпами роста величин t и x :

$$\frac{\Delta t}{t} \approx E_x(t) \frac{\Delta x}{x}.$$

Это означает, что если значение x (размерность задачи) повышается на один процент, то значение t (время выполнения алгоритма) увеличивается приблизительно на $E_x(t)$ процентов. В соответствии с данным физическим смыслом эластичности, теорема о классификации \mathcal{L} -функций порождает пять сложностных классов, которые полностью отвечают современной детальной классификации алгоритмов.

Класс быстрых алгоритмов — это множество алгоритмов, для которых функции сложности $t(x) \in \mathbf{Subpoly}^*$. Такие функции имеют тождественно нулевую или бесконечно малую эластичность, поэтому для быстрых алгоритмов время работы либо вообще не зависит от длины входа, либо *темп роста времени выполнения алгоритма асимптотически ниже темпа роста размерности решаемой задачи*.

Класс полиномиальных алгоритмов — это множество алгоритмов, для которых функции сложности принадлежат классу \mathbf{Poly} , а эластичность этих функций есть асимптотически постоянная величина. Для алгоритмов этого класса *темп роста времени их работы асимптотически соизмерим с темпом роста размерности задачи*.

Следует отметить, что классы быстрых и полиномиальных алгоритмов относятся к полиномиальным в классической систематизации. Их принято считать эффективными, практически значимыми, работающими “в реальном масштабе времени”. Поведение эластичности, присущее им, подтверждает и объясняет “реальный масштаб времени”.

Класс субэкспоненциальных алгоритмов — алгоритмы, для которых функции сложности входят в класс \mathbf{Subexp} . Это алгоритмы с более чем полиномиальной и менее чем экспоненциальной сложностью. Такие алгоритмы применяются для решения задач малой и средней размерности. Эластичность $E_x(t)$ субэкспоненциального алгоритма — бесконечно большая величина, порядок роста которой характеризуется отношениями $1 \prec E_x(t) \prec x$. Таким образом, для *субэкспоненциальных алгоритмов темп роста времени выполнения значительно выше темпа роста размерности задачи*. Между тем он все же ниже, чем

$$\frac{\Delta t}{t} \approx x \frac{\Delta x}{x} = \Delta x.$$

Класс экспоненциальных алгоритмов — это алгоритмы, для которых функции сложности $t(x) \in \mathbf{Exp}$. Для них эластичность $E_x(t) = O(x)$ — бесконечно большая величина, асимптотически пропорциональная линейной функции. Функции с подобной эластичностью описывают законы естественного роста — скорость увеличения такой функции прямо пропорциональна ей самой:

$$\frac{t'(x)}{t(x)} = \frac{E_x(t)}{x} = O(1).$$

Подобным законам подчиняются, например, рост числа бактерий, увеличение денежного вклада при постоянном процентном приращении, процесс радиоактивного распада и др. Все это стремительно протекающие процессы. *Время выполнения экспоненциальных алгоритмов растет так быстро, что на практике их можно использовать лишь для задач малой размерности.*

Класс гиперэкспоненциальных алгоритмов – это алгоритмы, с функциями сложности $t(x) \in \mathbf{Hyperexp}$. Для них $x \prec E_x(t)$. Темп роста гиперэкспоненциальных функций настолько высокий, что не укладывается в естественные законы природы. *В настоящее время алгоритмы класса **Hyperexp** практически неприменимы.* Только мощный прорыв в быстрой вычислительной технике (например, появление квантовых компьютеров или биокомпьютеров) позволит в будущем использовать такие алгоритмы при решении практических задач.

Классы субэкспоненциальных, экспоненциальных, гиперэкспоненциальных алгоритмов в классической систематизации составляют множество экспоненциальных алгоритмов. Эластичность функции сложности всякого алгоритма из данных классов — переменная бесконечно большая величина.

Совершенно неэластичные, эластичные и совершенно эластичные алгоритмы

Вновь обратимся к существующей аналогии между функциями сложности алгоритмов и производственными функциями. В экономических науках производственные функции с бесконечно большой положительной эластичностью называют совершенно эластичными. Например, для монотонно убывающей функции спроса $D = D(p)$, где p – цена некоторого товара, эластичность $E_p(D) = \infty$ означает, что самое малое снижение цены побуждает покупателя увеличивать покупки от нуля до предела своих возможностей. Совершенно неэластичный спрос $E_p(D) = 0$ задает другой крайний случай: изменение цены не приводит ни к какому изменению спроса. Средний случай, когда $|E_p(D)| = c > 0$ – константа, подразделяют на два: при $c > 1$ спрос считают эластичным, а при $c < 1$ – неэластичным [6, 8].

Поскольку функция сложности описывает обратную зависимость, нежели производственная функция, то применительно к сложностным классам алгоритмов уместна следующая терминология. Так, алгоритмы с функциями сложности из классов **Subexp**, **Exp**, **Hyperexp** целесообразно назвать *совершенно эластичными*, из класса **Poly** – просто *эластичными*, а быстрые алгоритмы – *совершенно неэластичными*. Если в экономических науках наибольший интерес вызывают эластичные и совершенно эластичные зависимости, то в метрической теории алгоритмов наоборот – совершенно неэластичные и эластичные алгоритмы.

Заключение

Предложенная детальная классификация алгоритмов на основе асимптотического поведения эластичности их функций сложности имеет ряд достоинств. Она не разрушает прежней, традиционной классификации с полиномиальными и экспоненциальными алгоритмами, а лишь дополняет и уточняет ее. Свойства эластичности позволяют без особого труда

вычислять ее для большинства функций сложности и определять сложностный класс исследуемого алгоритма. Возможно сравнение функций сложности, принадлежащих одному классу. Имеет место инвариантность введенных классов алгоритмов относительно модели вычислений и любых полиномиальных преобразований функций сложности. Допустимо дальнейшее развитие данной классификации в части использования эластичности функций многих переменных, введения классов алгоритмов, например, с полиномиальной, субэкспоненциальной и экспоненциальной эластичностью.

К ограничениям рассмотренной классификации следует отнести требование принадлежности функций сложности алгоритмов к семейству Харди. Однако в большинстве реальных случаев это требование является вполне естественным и не вызывает особых трудностей при анализе алгоритмов. Хочется надеяться, что широкая практика применения предложенной систематизации алгоритмов выявит другие ее плюсы и минусы.

Список литературы

- [1] В.В.Быкова, Т.П.Портнягина, Исследование угловой меры асимптотического роста функций сложности алгоритмов, *Труды Седьмой Всероссийской ФАМ'2008 конференции*, Ч. 2, Красноярск, СФУ, 2008, 47-57.
- [2] О.Н.Василенко, Теоретико-числовые алгоритмы в криптографии, М., МЦНМО, 2006.
- [3] В.А.Головешкин, М.В.Ульянов, Метод классификации вычислительных алгоритмов по сложности на основе угловой меры асимптотического роста функций, *Вычислительные технологии*, 11(2006), №1, 52-62.
- [4] Р.Грэхем, Д.Кнут, О.Поташник, Конкретная математика, М., Мир, Бином. Лаборатория знаний, 2006.
- [5] М.Гэри, Д.Джонсон, Вычислительные машины и труднорешаемые задачи, М., Мир, 1982.
- [6] К.Доугерти, Введение в эконометрику, М., ИНФРА-М, 2001.
- [7] Т.Кормен, Ч.Лейзерсон, Р.Ривест, Алгоритмы: построение и анализ, М., МЦНМО, 1999.
- [8] А.С.Солодовников, В.А.Бабайцев, А.В.Браилов, И.Г.Шандра. Математика в экономике, М., Финансы и статистика, 2001.
- [9] В.А.Успенский, А.Л.Семенов, Теория алгоритмов: основные открытия и приложения, М., Наука, 1987.
- [10] Г.Х.Харди, Курс чистой математики, М., ИЛ, 1949.
- [11] А.Л.Чмора, Современная прикладная криптография, М., Гелиос АРВ, 2001.
- [12] И.П.Натансон, Краткий курс высшей математики, СПб., Лань, 1997.

Recognition Method of Algorithms Classes on the Basis of Asymptotics for the Elasticity of Complexity Functions

Valentina V.Bykova

We offer a new indication to recognize the algorithms classes which is based on the asymptotic behavior of the elasticity of complexity functions. The present day analogy for functions of complexity algorithms and produced functions is used, the rate of which is traditionally evaluated by elasticity in econometrics. The theorem that states the characterization of elasticity for rapid, polynomial, subexponential, exponential and hyperexponential algorithms has been proved. The principal advantage of the suggested indication is that it allows the simplicity of computation caused by the well-known properties of elasticity.

Keywords: computation complexity, elasticity of algorithms.