

$P \stackrel{?}{=} NP$ или проблема перебора: взгляд из 90-х

А. А. Разборов
Математический Институт РАН
им. В.А.Стеклова

10 января 2001

1 Введение

$P \stackrel{?}{=} NP$ -проблема (или, как её иногда сейчас¹ неформально называют, *проблема перебора*) была впервые сформулирована в 1971-1972 гг., т.е. менее 30 лет назад. Несмотря на это, за сравнительно короткое время она успела превратиться в одну из центральных открытых задач современной математики, сравнимых по своей значимости с гипотезами Римана или Пуанкаре (ср. [Sma98]).

Данный краткий очерк в основном посвящён истории самой проблемы перебора: мы постараемся рассказать о развитии разнородных идей, приведших в конце концов к этому удивительному открытию. Читатель, интересующийся тем влиянием, которое $P \stackrel{?}{=} NP$ -проблема оказала на последующее развитие математики и информатики, а также историей многочисленных попыток её решения (включая работы автора данного очерка) найдёт для себя полезным обзор [Sip92]. Любопытные исторические замечания об исследованиях, проводимых в СССР в период, предшествовавший точной формулировке проблемы перебора, содержатся также в [Tra84, Sli99].

¹по поводу происхождения этого термина см. раздел 3.3

2 Предпрудыстория: теория алгоритмов и вычислимых функций

Вне всякого сомнения, наиболее важным понятием, лежащим в основе $P \stackrel{?}{=} NP$ -проблемы (как, впрочем, и одной из наиболее важных концепций, разработанных в математике XX века вообще) является фундаментальное определение *алгоритма* в строго математическом смысле этого слова. Более подробный рассказ об истории этого замечательного открытия, к сожалению, выходит далеко за рамки настоящего очерка. Поэтому мы ограничимся тем, что весьма бегло обрисует состояние теории алгоритмов (или, как её иногда называют, *теории вычислимых функций*) на начало 60-х годов, с какового периода, собственно, и начинается наша история.

К этому времени теория алгоритмов и вычислимых функций превратилась во вполне сложившуюся, стройную и бурно развивающуюся дисциплину. Одним из краеугольных камней этой теории служит так называемый *тезис Чёрча*, утверждающий, что все “разумные” вычислительные устройства эквивалентны между собой. Этот тезис позволяет раз и навсегда зафиксировать какой-либо один тип вычислительных устройств (в роли которого в силу исторических причин чаще всего выступают *машины Тьюринга*). Согласно тезису Чёрча, результаты, доказанные для машин Тьюринга, справедливы и для других вычислительных моделей и, стало быть, могут быть интерпретированы как результаты *об алгоритмах вообще*. К их числу, в частности, относятся яркие результаты об алгоритмической разрешимости или неразрешимости многих естественных (алгоритмических) задач алгебры и логики. И хотя всё ещё имеется ряд важнейших задач, для которых вопрос об алгоритмической разрешимости в настоящее время открыт, в целом созданные к концу 60-х годов мощные индустриальные методы и полученные к этому времени результаты можно рассматривать как сравнительно успешную классификацию алгоритмических задач в соответствии с их (алгоритмической) разрешимостью (см., например, обзор [Rab77]).

3 Предыстория: три составные части

Оглядываясь назад на приблизительно десятилетний период с начала 60-х до начала 70-х годов, непосредственно предшествовавший формулировке проблемы перебора в статьях Кука [Coo71b], Карпа [Kar72] и Левина [Лев73], можно, хотя и несколько условно, выделить три основных направления исследований, слившихся в конце концов в вышеупомянутых работах воедино:

- Абстрактная теория сложности вычислений
- Алгоритмы для конкретных комбинаторных задач
- Концепция перебора и его устранения.

До работ Кука-Карпа-Левина эти направления мало пересекались между собой, и мы также расскажем о них по отдельности.

3.1 Абстрактная теория сложности вычислений

Развитие теории сложности вычислений по существу началось с появлением первых ЭВМ. Специалисты начали естественно интересоваться не только вопросами о *существовании* алгоритма для той или иной алгоритмической задачи, но и тем, насколько построенные алгоритмы являются *эффективными*, т.е. осуществимыми с практической точки зрения. Основы общей теории эффективных алгоритмов и были заложены как раз в 60-е годы.

В самом начале этого периода выкристаллизовалось центральное понятие *сигнализирующей функции*² $T_M(x)$: это число тактов (т.е. время работы) машины Тьюринга (или любого другого вычислительного устройства – ср. раздел 2) на входных данных, представленных в виде двоичного слова x . В общем виде изучать эту функцию довольно трудно в силу её “рыхлости”: например, подавляющее большинство задач осмыслено лишь на малом числе входов x весьма специального вида, причём множество этих “осмысленных” входов для каждой задачи своё. По этой и другим причинам сравнить разумным образом сигнализирующие $T_M(x)$

²в данном кратком очерке нас интересуют только сигнализирующие по времени функции

и $T_{M'}(x)$ для двух различных алгоритмов M и M' удаётся лишь в исключительных случаях.

Следующей важнейшей концепцией, на которой основывается современная теория сложности вычислений, является понятие *сложности в наихудшем случае*, определяемой следующей простой формулой:

$$t_M(n) \stackrel{\text{def}}{=} \max_{|x| \leq n} T_M(x),$$

где $|x|$ — битовая длина двоичного слова x . Иными словами, сложность в наихудшем случае $t_M(n)$ *гарантирует* нам, что вычисление закончится за это время на *любом* входном слове x , коль скоро битовая длина последнего не превышает n . Оказывается, что на уровне функций $t_M(n)$ возможно уже вполне успешное построение разумной математической теории для сравнения сложности различных алгоритмов.

Концепция сложности в наихудшем случае, кажется, была известна в СССР уже в самом начале рассматриваемого периода (см., например, [Яно59] или подстрочное примечание на стр. 36 работы [Траб4]). Её распространение на Западе было слегка замедлено тем обстоятельством, что в англоязычной литературе первой половины 60-х гораздо чаще встречались так называемые *машины со входом* (в свою очередь возникшие как обобщение крайне популярных в то время конечных автоматов), для которых естественного аналога сложности в наихудшем случае не существует. По-видимому, сложностные аспекты вычислений на обыкновенных³ машинах Тьюринга были впервые в англоязычной литературе рассмотрены в [Неп65]; в той же работе появилось и современное определение сложности в наихудшем случае, приведённое выше.

Пусть теперь имеется алгоритмическая задача, записанная в виде задачи вычисления отображения $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ множества конечных двоичных слов в себя. Если бы для каждой вычислимой (некоторым алгоритмом) функции f существовал вычисляющий её алгоритм M , оптимальный с точки зрения асимптотического поведения сложности в наихудшем случае $t_M(n)$, то дальнейшее построение теории сложности можно было бы существенно упростить (а именно, объявив $t_M(n)$ для такого оптимального алгоритма M сложностью самой функции f). Доказанная в [Blu67] *теорема об ускорении* убедительно продемонстрировала, что

³т.е. вычисляющих функции из конечных слов в конечные слова — в настоящее время эта модель является общепринятой

этим надеждам сбыться не суждено: существуют вычислимые функции f , для которых любой алгоритм её вычисления может быть произвольно (скажем, в “логарифм раз”) ускорен.

Альтернативой этому наивному подходу является другое центральное понятие теории сложности вычислений, а именно понятие *класса сложности*. В общем виде классы сложности определяются так: фиксируется множество всех алгоритмов определённой формы (которая, как мы увидим ниже, может быть весьма разнообразной и не обязательно соответствовать “реальным” вычислениям) и удовлетворяющих тем или иным сложностным ограничениям. После этого в рассматриваемый класс зачисляются все алгоритмические задачи, для которых *существует хотя бы один* алгоритм из этого множества.

В таком виде понятие класса сложности встречается уже в ранней работе [Rit63], однако в ней рассматриваются лишь ограничения на память, используемую алгоритмами довольно специального вида. Наиболее важные для построения теории сложности классы

$$TIME(t(n)) \stackrel{\text{def}}{=} \{ f \mid \exists M (M \text{ вычисляет } f \ \& \ \forall n (t_M(n) \leq t(n))) \}$$

впервые появляются в основополагающей работе [HS65] в контексте упомянутых выше машин со входом. *Теорема об иерархии*, являющаяся краеугольным камнем современной теории сложности вычислений, утверждает, что если $t_2(n)$ хотя бы “слегка больше” чем $t_1(n)$, то класс $TIME(t_2(n))$ содержит функцию f , не принадлежащую $TIME(t_1(n))$. Иными словами, иерархия классов сложности $TIME(t(n))$ собственная — например, все включения в цепочке

$$TIME(n) \subsetneq TIME(n^2) \subsetneq TIME(n^3) \subsetneq \dots \subsetneq TIME(2^n)$$

строгие. Эта теорема была доказана в [HS65] для машин со входом и затем распространена (и при этом слегка усилена) на случай обыкновенных машин в [HS66] (см. также [Яно59] и обсуждение в разделе 3.3).

Наконец, ещё одним замечательным открытием теории сложности, сделанным до наступления **NP**-эры⁴ стала формулировка класса слож-

⁴Следуя [Tra84], так мы будем называть период, наступивший после работ Кука-Карпа-Левина. Отчётливо осознавая некоторую патетичность этого термина, хочется всё же отметить, что данные работы повлекли за собой коренное изменение ориентиров и идеологии в нескольких важнейших смежных дисциплинах. Поэтому в данном случае его употребление нам кажется вполне оправданным.

ности $\mathbf{P}$ задач, разрешимых за полиномиальное время:

$$\mathbf{P} \stackrel{\text{def}}{=} \bigcup_{k \geq 0} \text{TIME}(n^k)$$

и осознание его важности. Этот класс был в явном виде впервые определён Кобхэмом в работе [Cob64] (к сожалению, оказавшейся в своё время несколько в тени, по крайней мере, среди специалистов по теории сложности) и фигурирует в обзоре Рабина [Rab66] по теории автоматов (см. также обсуждение работ Эдмондса в следующем разделе 3.2). Забегая несколько вперёд, следует отметить, что в настоящее время класс $\mathbf{P}$ рассматривается как общепринятое теоретическое приближение к классу задач, на самом деле поддающихся решению на реально существующих компьютерах (развёрнутое обсуждение этого вопроса можно найти в пленарном докладе Кука на международном конгрессе математиков 1990 г. в Киото [Coo90]). Однако подлинное осмысление этого тезиса стало возможным лишь после слияния абстрактной теории сложности вычислений с теорией алгоритмов для конкретных комбинаторных задач, ознаменовавшего собой наступление $\mathbf{NP}$ -эры (см. раздел 4.2 ниже). В ранних работах [Cob64, Rab66] он отмечается всё ещё в предельно осторожной форме и с массой оговорок.

Таким было в самых общих чертах состояние теории сложности вычислений до работ Кука, Карпа и Левина. В то время как теоретические основы классификации алгоритмических задач в соответствии с их внутренней сложностью были в основном заложены, начинал ощущаться острый дефицит интересных задач, к которым построенная теория могла бы быть приложена. Мы завершаем данный раздел характерной для того времени цитатой из одной из последних работ [Coo71a], непосредственно предшествовавших наступлению $\mathbf{NP}$ -эры:

И, наконец, в области вычислительной сложности давно стоит проблема нахождения какого-нибудь интересного множества (двоичных слов, т.е. алгоритмической задачи распознавания – прим. авт.) и доказательства того, что для его вычисления всякая разумная модель вычислителя потребует много времени. Более точно, не удаётся найти множество, не содержащееся в классе Кобхэма $\mathcal{L}_$ ($= \mathbf{P}$ – прим. авт.) (не считая примеров, построенных с помощью диагональной процедуры).*

3.2 Алгоритмы для комбинаторных задач

Между тем, большое количество интересных задач, к которым теория сложности могла бы быть потенциально приложена, рассматривалось, одновременно и фактически независимо, в смежной области построения конкретных алгоритмов для вполне конкретных комбинаторных (чаще всего, оптимизационных) задач на графах, матроидах, потоковых задач, задач минимизации булевых функций и т.д. Важнейшей отличительной чертой возникавших задач было то, что существование *какого-либо* алгоритма для большинства из них было вполне очевидно, и таким образом эти задачи не представляли никакого интереса для рассмотренной нами в разделе 2 классической теории вычислимых функций. Эта область также была вызвана к жизни появлением первых ЭВМ (стимулировавшим интерес именно к *эффективным* алгоритмам), и на первоначальных этапах её развития разработчиками комбинаторных алгоритмов на эмпирическом уровне были независимо нащупаны многие важные понятия, легшие впоследствии в основу общей стройной теории.

Вполне отчётливое понимание разницы между “механической процедурой” (= алгоритмом – *прим. авт.*) и “механической процедурой, достаточно короткой для того, чтобы быть практичной” прослеживается уже, например, в ранней работе [Qui52], посвящённой минимизации булевых функций. На примере рецензии [Nel55] на эту статью хорошо видно какие оживлённые (и в данном случае довольно персонифицированные) дискуссии могли в отсутствие предоставляемых теорией сложности чётких ориентиров вызывать вопросы о том, является ли тот или иной алгоритм “практическим” или “чисто механическим”.

Положение вещей, при котором специалисты по решению комбинаторных задач довольствовались интуитивными представлениями о том, что такое “практический” алгоритм сохранялось до работ Эдмондса [Edm65b, Edm65a]. Однако уже раздел 2 работы [Edm65b], озаглавленный “Digression”, целиком посвящён обсуждению концепции полиномиального алгоритма и, в такой же осторожной форме, как и в отмеченных выше работах [Cob64, Rab66], содержит тезис о том, что алгоритмы работающие за полиномиальное время с известными оговорками могут быть отождествлены с “практическими” алгоритмами.

В любом случае, подавляющее большинство задач, с которыми в то время столкнулись исследователи, не поддавались решению с помощью “практических” алгоритмов ни в каком разумном смысле этого слова.

В попытках понять, хотя бы на интуитивном уровне, какие же из этих задач на самом деле более трудны чем остальные, сформировалось другое центральное понятие, впоследствии лёгшее в основу общей теории – понятие *сводимости*. Пусть у нас есть пара алгоритмических задач $f_1, f_2 : \{0, 1\}^* \rightarrow \{0, 1\}^*$ и некоторая функция g , *сводящая* f_1 к f_2 , т.е. такая, что $\forall x (f_1(x) = f_2(g(x)))$. Тем самым, если сводящая функция g легко вычислима (например, лежит в классе **P** – в этом случае сводимость называют *полиномиальной*), то задача f_1 заведомо не сложнее задачи f_2 : всякий эффективный алгоритм для вычисления f_2 может быть очевидным образом преобразован в эффективный же алгоритм для вычисления f_1 .

Общее понятие (полиномиальной) сводимости, являющееся одним из краеугольных камней современной теории сложности, было впервые предложено в связи с формулировкой $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ проблемы в работах Кука-Карпа-Левина. В рассматриваемый нами сейчас период времени в литературе начали накапливаться лишь первые частные примеры сводимостей (см., например, [Dan60, Edm62, Gim65]).

Подытоживая вышесказанное, до появления теории **NP**-полноты в начале 70-х, в области построения комбинаторных алгоритмов был предложен ряд эффективных алгоритмов для нескольких важных комбинаторных задач и, параллельно с абстрактной теорией сложности, в осторожной форме был высказан тезис о практичности полиномиальных алгоритмов и о полиномиальности практичных. Существенно бóльшая часть задач не поддавалась эффективному решению, и для таких задач стали появляться отдельные результаты об их сводимости друг к другу. Однако, отсутствие в то время строгих сложностных концепций “переборной” задачи и полиномиальной сводимости не позволило продвинуться в построении классификации “предположительно сложных” комбинаторных задач далее изолированных примеров.

3.3 Концепция перебора и его устранения

Наконец, ещё одно направление, которое следует отметить в нашем кратком очерке предыстории $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ -проблемы – это исследования, проводимые главным образом в СССР и специально сфокусированные на понимании феномена перебора (кстати, сам термин “проблема перебора” восходит именно к этому периоду, хотя, разумеется, в то время он понимался

совершенно неформально). В школе С. В. Яблонского (в организации и работе которой на ранних этапах принимал участие также А. А. Ляпунов) рассматривались преимущественно “кибернетические” задачи минимизации схем различного вида и так называемых дизъюнктивных нормальных форм (ДНФ) булевых функций, построения сложных булевых функций и др. В процессе этой деятельности была довольно устойчиво выработана важнейшая концепция, согласно которой трудность алгоритмических задач определяется именно “количеством” заложенного в них перебора вариантов.

Автор данного очерка начал свою научную карьеру намного позднее рассматриваемого периода и при его подготовке столкнулся в этом месте с существенными трудностями. Многие идеи и даже результаты, полученные в это время в СССР, существовали лишь в форме устного творчества и никогда не были опубликованы. О том, насколько замечательны были некоторые из них, можно достоверно судить по одному из немногих дошедших до нас письменных источников той эпохи – конспекту доклада Г. С. Цейтина, сделанного в 1956 г. на двух заседаниях семинара МГУ по математической логике [Яно59]. Согласно этому источнику, в СССР уже тогда были известны понятия сложности в наихудшем случае и теорема об иерархии, по крайней мере в контексте нормальных алгорифмов Маркова.

Тем не менее, настоящий очерк называется “взгляд из 90-х”, и безусловное предпочтение в нём отдаётся тем работам и идеям, непрерывное и систематическое развитие которых привело к становлению теории сложности вычислений (и теории **NP**-полноты в частности) в её современном виде. К сожалению, роль блестящих, но изолированных, неоформленных и неопубликованных идей в этом контексте весьма ограничена: как правило, они живут ровно столько, сколько они интересны своим непосредственным создателям. Поэтому автор решил намеренно преодолеть искушение провести систематическое исследование на тему о том, родиной какой именно части современной теории сложности является СССР, и ограничиться общедоступными из литературы фактами.

Важной вехой рассматриваемого периода стали работы С. В. Яблонского [Ябл59a, Ябл59b], в которых для задачи построения сложной булевой функции строго доказано отсутствие эффективных алгоритмов в довольно специальном классе так называемых правильных алгоритмов. Ввиду значительного несоответствия ценности полученных результатов приписываемой им интерпретации, эти работы впоследствии получили

весьма неоднозначную оценку (ср. [Tra84]). С одной стороны, сам Яблонский после выхода в свет его работ считал проблему перебора закрытой и свободно использовал имевшиеся в его распоряжении административные возможности для “убеждения” в этом научного сообщества. С другой стороны, *de facto* эти работы сыграли и важную положительную роль в привлечении к данной тематике внимания других исследователей.

Для отечественных статей этого периода характерен интерес к частным классам алгоритмов, определяемых в соответствии со спецификой рассматриваемой задачи. К сожалению, из-за недооценки важности общих унифицирующих понятий теории вычислимости и теории сложности (проявившейся, помимо прочего, в отмеченном выше пренебрежении созданием систематической теории на текстуальном уровне), в современной форме проблема перебора могла быть сформулирована лишь в работе Левина [Лев73], хотя все остальные необходимые составляющие, кажется, были осознаны уже много ранее.

4 История: три источника

В предыдущей главе мы попытались передать то предгрозовое ощущение, которое испытывали специалисты в нескольких соседних “околокомпьютерных” областях к концу 60-х годов. Было ясно, что назревает нечто весьма серьёзное.

“Гром грянул” в работе Кука [Coo71b], и этот гром немедленно перешёл в грозу в статье Карпа [Kar72] (в силу очевидных исторических причин, работа [Лев73], в которой независимо были получены многие результаты из [Coo71b, Kar72], стояла несколько особняком). Рассказывая об этих основополагающих работах, мы постараемся также хотя бы вкратце коснуться тех плодов, которые к настоящему времени принесли заложенные в них идеи.

4.1 Кук: Сложность процедур вывода теорем

В этой работе впервые рассмотрен класс сложности **NP**, состоящий из тех задач, которые можно решить за полиномиальное время на *недетерминированной* машине Тьюринга; в настоящее время этот класс рассматривается как строгий математический эквивалент неформальной концепции “переборной” задачи.

Недетерминированная машина Тьюринга (НМТ) – это машина, которая специально приспособлена для распознавания языков, т.е. для решения алгоритмических задач, ответ на которые имеет вид ДА/НЕТ. В ходе своей работы НМТ может написать в какой-то ячейке знак вопроса, после чего её дальнейшая работа “недетерминированно разделяется” на две ветви, в зависимости от того, заменяется ли ? на 0 или 1. В результате вычисление НМТ оказывается двоичным деревом, вообще говоря, экспоненциального размера. Вычисление считается успешным на входе x (т.е. $f(x) = \text{ДА}$), если вдоль хотя бы одной ветви был получен ответ ДА, и неуспешным ($f(x) = \text{НЕТ}$) в противном случае.

Следует отметить, что на современном уровне развития естествознания НМТ являются чисто воображаемой вычислительной моделью, и неизвестен никакой способ их эффективного моделирования с помощью существующей в реальности аналоговой модели. По-видимому, наиболее перспективными исследованиями в этом направлении являются попытки моделирования недетерминированного ветвления с помощью молекул ДНК (см., например, обзор [BF98]), однако даже они, кажется, пока довольно далеки от своего логического завершения.

Недетерминированные машины Тьюринга были известны в литературе и до статьи [Coo71b]. Тем не менее, именно в этой работе они впервые получают постоянную прописку в качестве полноправного объекта исследований, а не некоторой курьёзной патологии. Тем самым был создан важнейший прецедент рассмотрения класса сложности, соответствующего вымышленной, не существующей в реальном мире вычислительной модели. Такой подход оказался удивительно плодотворным для дальнейшего развития теории сложности, и многие важнейшие достижения последних лет были получены в рамках именно этой схемы. Чтобы дать хотя бы общее представление, о чём в данном контексте идёт речь, достаточно упомянуть *интерактивные доказательства* (премия им. Неванлинны за 1994 г. – А. Wigderson), *вероятностно проверяемые доказательства* (PCP) и *квантовые вычисления* (премия им. Неванлинны за 1998 г. – Р. Shor). К этим моделям мы ещё вернёмся в следующем разделе 4.2.

Другим важнейшим понятием, рассмотренным в работе [Coo71b], является понятие *полиномиальной сводимости*. Мы уже видели в разделе 3.2, как эта концепция в течение предшествующего периода вызревала в недрах теории комбинаторных алгоритмов. Что же касается сегодняш-

него дня, современная теория сложности вычислений просто немыслима без общего понятия сводимости. Буквально каждый раздел этой дисциплины цементируется по крайней мере одним понятием сводимости, специфическим для этого раздела. Именно понятие сводимости превращает теорию сложности вычислений из набора разрозненных эмпирических фактов в строгую и элегантную математическую теорию.

Собственно говоря, в самом по себе понятии сводимости нет ничего глубокого и удивительного. Оно является стандартным рабочим инструментом уже в теории вычислимых функций (отмеченные в разделе 2 “индустриальные” методы доказательства неразрешимости как раз большей частью и состоят в построении подходящих сводимостей), а с математической точки зрения суть не что иное как морфизмы в некоторых очевидных категориях, естественно приспособленных для изучения полиномиальных вычислений. Полностью изменить облик целой дисциплины понятию сводимости позволил лишь тот эмпирический факт, что **число сводимостей одних важных задач к другим огромно**, и, что ещё важнее, **существуют стандартные методы построения сводимостей**, основанные на введении в рассмотрение воображаемых вычислительных моделей и возникающих классов сложности (см., например, [GJ79, гл. 3,7]). И снова в работе [Coo71b] был сделан первый важнейший шаг в этом направлении: было показано, что *любая* задача из класса **NP** полиномиально сводится к вполне определённой задаче **ВЫПОЛНИМОСТЬ** (пропозициональных формул) из этого класса с помощью некоторого общего метода, основанного на анализе соответствующего недетерминированного вычисления. Тем самым **ВЫПОЛНИМОСТЬ** является “самой сложной” задачей в классе **NP**; общепринятый в настоящее время термин для таких задач – **NP-полная задача**. Такая схема классификации оказалась удивительно плодотворной: в подавляющем большинстве случаев естественные алгоритмические задачи оказываются полны в том или ином естественном же классе сложности.

Наконец, в [Coo71b] содержится первый пример доказательства **NP**-полноты для *чисто комбинаторной* задачи **ИЗОМОРФИЗМ ПОДГРАФУ**. Последний результат послужил прелюдией к следующей важной вехе в развитии интересующего нас предмета, а именно, работе Карпа [Kar72], к изложению которой мы сейчас и переходим.

4.2 Карп: Сводимость комбинаторных проблем

С моей точки зрения, наиболее удивительным обстоятельством, связанным с этой работой, является то, что практически вся терминология (и даже обозначения!), а также все предложенные в ней концепции оказались настолько удачными, что они глубоко укоренились в теории сложности и используются по сей день практически без изменений. В этом смысле статья [Kar72] стала первой убедительной реализацией в виде стройной теории идей, намеченных в [Coo71b].

Карп ввёл обозначения $\mathbf{P}$ и $\mathbf{NP}$. Он также предложил свой вариант полиномиальной сводимости (известной сейчас как *сводимость по Карпу*); как самой этой сводимости, так и введённому для неё обозначению $\propto$ было суждено вскоре стать общеупотребимыми. Неоднократно упомянутый выше тезис о том, что понятие полиномиального алгоритма является удовлетворительным приближением к понятию “практического” алгоритма во вполне уверенной и определённой форме выдвигается именно в работе [Kar72]. Здесь же начато систематическое изучение класса $\mathbf{P}$ задач, разрешимых с помощью полиномиальных алгоритмов.

Именно в статье Карпа впервые вводится термин $\mathbf{NP}$ -полная задача, в явном виде формулируется $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ -проблема и в явном же виде отмечается причина важности полных задач: *Очевидно, что либо все полные языки принадлежат $\mathbf{P}$, либо ни один из них не принадлежит $\mathbf{P}$. Первое имеет место тогда и только тогда, когда $\mathbf{P} = \mathbf{NP}$.*

Переходя от методологической ценности работы Карпа к доказанным в ней конкретным результатам, следует в первую очередь вспомнить в этой связи общепринятый на сегодняшний день термин “список Карпа из 21 полной задачи”. Мы уже говорили о том, что главная причина успеха современной теории сложности вычислений (и, в частности, схемы исследований проторенной проблемой перебора) состоит в том эмпирическом факте, что существует очень много разнообразных сводимостей между естественными алгоритмическими задачами. Как мы уже отмечали выше в разделе 3.2, в предшествовавшей литературе по комбинаторным алгоритмам встречались лишь частные примеры сводимостей (и не было никакого общего определения); напротив, статья Кука [Coo71b] является строго математической, однако в ней в основном рассматривается задача ВЫПОЛНИМОСТЬ, представляющая лишь ограниченный интерес для разработчиков практических алгоритмов. Начиная с работы [Kar72], этим направлениям было суждено слиться воедино. В ней была доказана

NP-полнота большого числа (а именно, 21 – прим. авт.) классических трудных вычислительных проблем, возникающих в таких областях как математическое программирование, теория графов, комбинаторика, вычислительная логика, теория переключательных схем. Тем самым было убедительно показано, что абстрактные концепции теории сложности вычислений (и, в частности, концепции, предложенные в [Coo71b]) замечательно прилагаются к вполне практическим и осязаемым вещам, а стремление к математической строгости и элегантности может прекрасно уживаться с той интуитивной и приходящей из практики интерпретацией, которую в любом случае желали бы сохранить специалисты по комбинаторным алгоритмам. Уже через несколько лет список **NP**-полных задач насчитывал тысячи задач, и в настоящее время тест на **NP**-полноту является обязательным для любой переборной задачи, для которой первые попытки построения эффективного алгоритма не приводят к успеху.

Статья Карпа по существу заложила основы методологии, которая является основой современной теории сложности вычислений и одновременно её наиболее характерной отличительной чертой по сравнению с другими разделами математики. Удивительным образом наиболее абстрактные, вымышленные модели “вычислений” оказываются *непосредственно* ответственными за внутреннюю сложность огромного числа вполне конкретных практических задач. Из числа наиболее ярких достижений последних лет в этом направлении можно отметить следующие:

- применение теории интерактивных доказательств и вероятностно проверяемых доказательств к классификации оптимизационных задач (см., например, обзор [Aro94]);
- применение квантовых вычислений к классификации задач, лежащих в основе современной практической криптографии (таких как ФАКТОРИЗАЦИЯ и ДИСКРЕТНЫЙ ЛОГАРИФМ [Sho97]).

4.3 Левин: универсальные задачи перебора

Работа [Лев73], независимо от Кука и Карпа, во вполне явном виде содержит понятие полиномиального алгоритма (“алгоритм, время работы которого сравнимо с длиной входа” в терминологии [Лев73]), задачи из класса **NP** (“квазипереборная задача”) и **NP**-полная задача (“универсальная переборная задача”). Кроме того, [Лев73] содержит список,

состоящий из шести естественных **NP**-полных задач, включающий **ВЫПОЛНИМОСТЬ И ИЗОМОРФИЗМ ПОДГРАФУ**, и утверждение об их **NP**-полноте (без доказательства). Тем самым, в работе Левина (в характерной для этого автора предельно лаконичной форме) были независимо сделаны многие важнейшие открытия как из [Coo71b], так и из [Kar72].

Помимо этого, [Лев73] содержит оригинальное определение сводимости, известной в настоящее время как *сводимость по Левину*, потенциальные преимущества которого (по сравнению с более стандартными сводимостями по Куку и Карпу) начали осознаваться лишь сравнительно недавно (см., например, [Aro95]). Другим важным результатом, доказанным в [Лев73], является теорема о существовании оптимального (в некотором смысле) алгоритма для любой задачи из класса **NP**: ожидаемый а priori аналог теоремы Блюма об ускорении [Blu67] для переборных задач на самом деле места не имеет.

5 Заключение

Одна из центральных открытых проблем современной математики, $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ -проблема, удивительно многолика, и, с моей точки зрения, стремительный взлёт её популярности за последние несколько десятилетий объясняется именно этим. С одной стороны, она обладает предельно естественной, ясно формулируемой и весьма элегантно строгой математической постановкой. С другой стороны, эта проблема имеет непосредственное отношение к огромному количеству вполне реальных задач, возникающих практически в любой области человеческой деятельности, в которой только используются методы математики и информатики. Наконец, особое очарование $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ -проблеме придаёт то обстоятельство, что это по существу строгая математическая формулировка извечного общепhilosophического вопроса о возможности элиминации (или хотя бы уменьшения количества) грубой силы (т.е. перебора).

В настоящем очерке мы постарались рассказать об истории этого удивительного открытия и объяснить одну из причин такой многоликости, проследив за тем как в $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ -проблеме сплелись воедино несколько направлений исследований в различных (хотя и смежных) областях, до этого проводимых разными группами в разных странах. К сожалению, мы смогли лишь бегло коснуться влияния, оказанного и оказываемого проблемой перебора на состояние современной теории сложности вычис-

лений.

И совсем за кадром остались непрекращавшиеся с 1971 г. попытки получить решение самой проблемы перебора. Хотя, как известно, на момент написания данного очерка такое решение и не получено, доказанные в этом направлении частные результаты, а также развитые в процессе их получения концепции и методы уже оказали на теорию сложности влияние, вполне сопоставимое с влиянием, оказанном самой $P \stackrel{?}{=} NP$ -проблемой (см., например, [Sip92]).

Сейчас трудно сказать будут ли в обозримом будущем найдены кардинально новые идеи и подходы, по-видимому, необходимые для окончательного решения проблемы перебора. Однако в любом случае либо самой этой проблеме, либо тем глубоким концепциям и идеям как уже вызванным ею к жизни, так и тем которые вне всякого сомнения в изобилии последуют из её гипотетического решения, суждено стать одним из наиболее ярких ориентиров в математике следующего столетия.

6 Благодарности

Я глубоко признателен М. В. Вялому, Л. А. Левину и А. О. Слисенко за ряд полезных замечаний, высказанных после ознакомления с предварительным вариантом текста. Написание этой статьи вряд ли было бы возможно без терпения и понимания, проявленного во время работы над ней моей супругой Ирэн; помимо этого, я благодарен ей за ряд конкретных стилистических исправлений.

Литература

- [Aro94] S. Arora. *Probabilistic Checking of Proofs and Hardness of Approximation Problems*. Electronic Colloquium on Computational Complexity, http://www.eccc.uni-trier.de/eccc-local/ECCC-Books/sanjeev_book_readme.html, 1994.
- [Aro95] S. Arora. Reductions, codes, PCPs and inapproximability. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 404–413, 1995.

- [BF98] R. Beigel and Bin Fu. Solving intractable problems with DNA computing. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity*, pages 154–168, 1998.
- [Blu67] M. Blum. A machine-independent theory of the complexity of recursive functions. *Journal of the ACM*, 14:322–336, 1967. Рус. пер.: М. Блюм, Машинно-независимая теория сложности рекурсивных функций, в сб. *Проблемы математической логики*, М.: Мир, 1970, стр. 401-422.
- [Cob64] A. Cobham. The intrinsic computational difficulty of functions. In *Proc. of the 1964 International Congress for Logic, Methodology, and the Philosophy of Science*, pages 24–30, Amsterdam, 1964. North Holland.
- [Coo71a] S. A. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *Journal of the ACM*, 18(1):4–18, 1971. Рус. пер.: С. А. Кук, Характеристики магазинных автоматов в терминах вычислителей, ограниченных во времени, в сб. *Сложность вычислений и алгоритмов*, М.: Мир, 1974, стр. 266-288.
- [Coo71b] S. A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971. Рус. пер.: С. А. Кук, Сложность процедур вывода теорем, *Киб. сборник (новая серия)*, вып. 12, М.: Мир, 1975, стр. 5-15.
- [Coo90] S. A. Cook. Computational complexity of higher type functions. In *Proceedings of the ICM90*, pages 55–69, 1990. Рус. пер.: С. А. Кук, Вычислительная сложность функций высшего типа, *Международный конгресс математиков в Киото*, М.: Мир, 1996, стр. 78-100.
- [Dan60] G. B. Dantzig. On the significance of solving linear programming problems with some integer variables. *Econometrica*, 28(1):31–44, January 1960.
- [Edm62] J. Edmonds. Covers and packings in a family of sets. *Bull. American Math. Soc.*, 68:494–499, 1962.

- [Edm65a] J. Edmonds. Minimum partition of a matroid into independent sets. *Journal of Research of the National Bureau of Standards (B)*, 69:67–72, 1965.
- [Edm65b] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
- [Gim65] J. F. Gimpel. A method of producing a Boolean function having an arbitrarily prescribed implicant table. *IEEE Transaction on Computers*, 14:485–488, 1965.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability. A guide to the theory of NP-completeness*. W. H. Freeman, 1979. Рус. пер.: М. Гэри, Д. Джонсон, *Вычислительные машины и труднорешаемые задачи*, М.: Мир, 1982.
- [Hen65] F. C. Hennie. One-tape, off-line turing machine computations. *Information and Control*, 8(6):553–578, 1965. Рус. пер.: Ф. К. Хенни, Вычисления на одноленточной машине Тьюринга с записью на ленте, в сб. *Проблемы математической логики*, М.: Мир, 1970, стр. 223–248.
- [HS65] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Trans. Amer. Math. Soc.*, 117:285–306, 1965. Рус. пер.: Дж. Хартманис, Р. Е. Стирнз, О вычислительной сложности алгоритмов, *Киб. сборник*, вып. 4, М.: Мир, 1967, стр. 57–85.
- [HS66] F. C. Hennie and R. E. Stearns. Two-tape simulation of multitape turing machines. *Journal of the ACM*, 13(4):533–546, 1966. Рус. пер.: Ф. К. Хенни, Р. Е. Стирнз, Моделирование многоленточной машины Тьюринга на двуленточной, в сб. *Проблемы математической логики*, М.: Мир, 1970, стр. 194–212.
- [Kar72] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. Рус. пер.: Р. М. Карп, Сводимость комбинаторных проблем, *Киб. сборник (новая серия)*, вып. 12, М.: Мир, 1975, стр. 16–38.

- [Nel55] R. J. Nelson. Review of [Qui52]. *J. Symbolic Logic*, 20:105–108, 1955.
- [Qui52] W. V. Quine. The problem of simplifying truth functions. *The American Mathematical Monthly*, 59:521–531, 1952.
- [Rab66] M. Rabin. Mathematical theory of automata. In *Proceedings of 19th ACM Symposium in Applied Mathematics*, pages 153–175, 1966.
- [Rab77] M. Rabin. Decidable theories. In J. Barwise, editor, *Handbook of Mathematical Logic*, chapter C.3. North-Holland, 1977. Рус. пер. в книге *Справочная книга по математической логике, часть III (теория рекурсии)*, М.: Наука, 1982.
- [Rit63] R. W. Ritchie. Classes of predictably computable functions. *Trans. Amer. Math. Soc.*, 106:139–173, 1963. Рус. пер.: Ричи Р.В., Классы предсказуемо вычислимых функций, в сб. *Проблемы математической логики*, М.: Мир, 1970, стр. 50-93.
- [Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.
- [Sip92] M. Sipser. The history and status of the P versus NP problem. In *Proceedings of the 24th ACM Symposium on Theory of Computing*, pages 603–618, 1992.
- [Sli99] A. Slissenko. Leningrad/st. petersburg (1961-1998): From logic to complexity and further. In C. S. Calude, editor, *People and Ideas in Theoretical Computer Science*, pages 274–313. Springer-Verlag, 1999.
- [Sma98] S. Smale. Mathematical problems for the next century. *Math. Intelligencer*, 20(2):7–15, 1998.
- [Tra84] B. A. Trakhtenbrot. A survey of Russian approaches to *perebor* (brute-force search) algorithms. *Annals of the History of Computing*, 6(4):384–400, October 1984.

- [Лев73] Л. А. Левин. Универсальные проблемы перебора. *Проблемы передачи информации*, 9(3):115–116, 1973. Engl. transl.: L. Levin, Universal sequential search problems, *Probl. Information Transmission* 9(1973), No 3, pp. 265-266.
- [Тра64] Б. А. Трахтенброт. Тьюринговы вычисления с логарифмическим замедлением. *Алгебра и логика*, 3(4):33–48, 1964.
- [Ябл59а] С. В. Яблонский. О невозможности элиминации перебора всех функций из P_2 при решении некоторых задач теории схем. *ДАН СССР*, 124(1):44–47, 1959.
- [Ябл59b] С. В. Яблонский. Об алгоритмических трудностях синтеза минимальных контактных схем. В сб. *Проблемы кибернетики, вып. 2* (ред. А. А. Ляпунов), стр. 75–121. Физматгиз, Москва, 1959. Engl. transl.: *Problems of Cybernetics*, Pergamon Press, 401-457, 1961.
- [Яно59] С. А. Яновская. Математическая логика и основания математики. В сб. *Математика в СССР за 40 лет. 1917–1957, т.1*, стр. 13–120. Физматгиз, Москва, 1959 (стр. 44-45: работы Г. С. Цейтина).