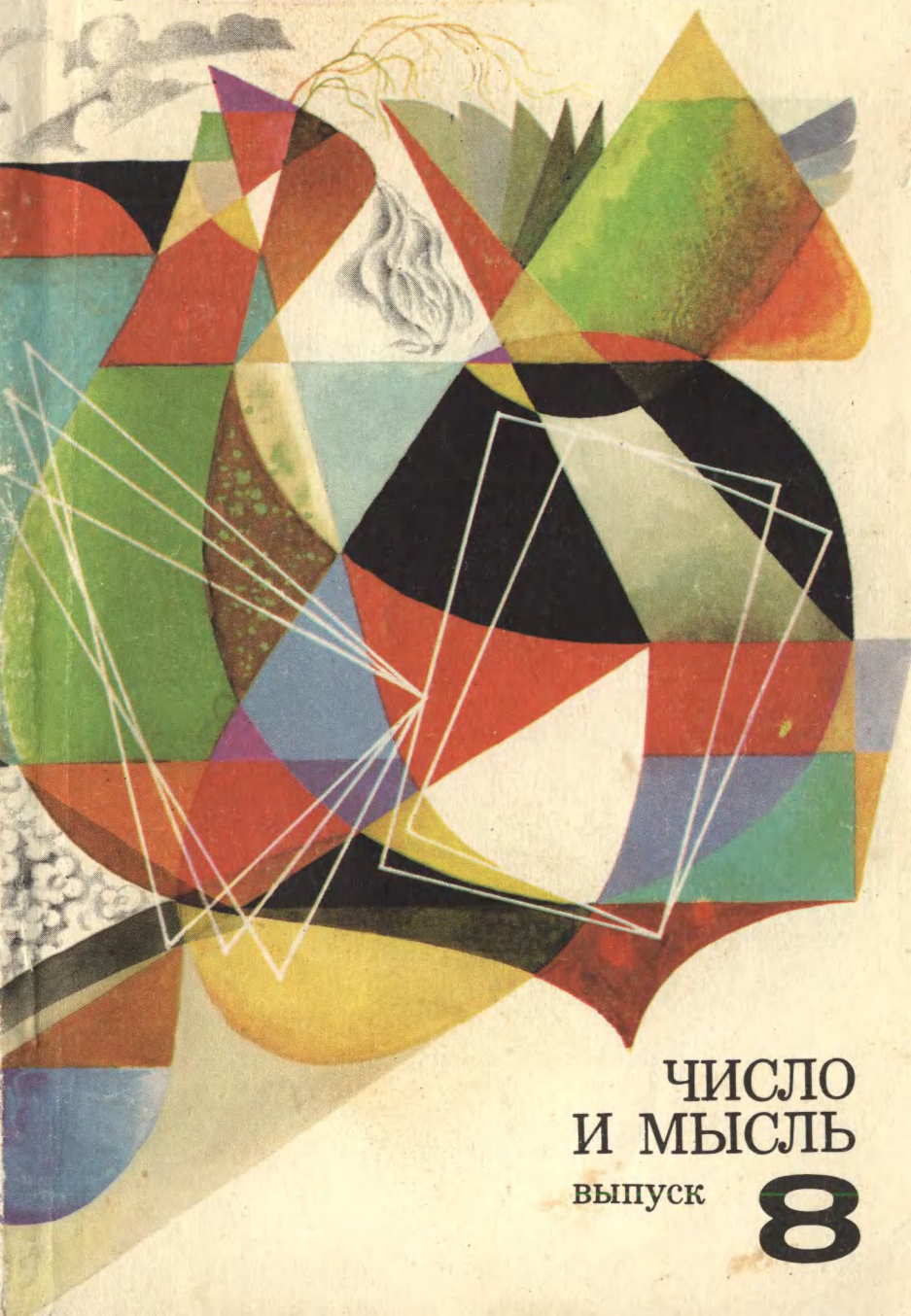


ЗНАНИЕ

НАРОДНЫЙ УНИВЕРСИТЕТ • 1985
естественнонаучный факультет



ЧИСЛО
И МЫСЛЬ
выпуск

8

Народный университет
Естественнонаучный факультет

Издается с 1961 года

Д. Б. ЮДИН,
доктор технических наук

А. Д. ЮДИН,
кандидат экономических наук



ЧИСЛО И МЫСЛЬ

выпуск 8

(Математики измеряют сложность)

ИЗДАТЕЛЬСТВО «ЗНАНИЕ» Москва 1985

ББК 65.9(2) 23

Ю 16

ЮДИН Давид Борисович — доктор технических наук, профессор МГУ имени Ломоносова, автор 15 монографий и многих работ по математическим методам планирования, управления и проектирования, лауреат Международной премии по дискретной математике.

ЮДИН Александр Давидович — кандидат экономических наук, старший научный сотрудник ВНИИКС, автор ряда работ по методам обработки статистической информации.

Рецензент: доктор философских наук **Б. В. Бирюков**.

Юдин Д. Б., Юдин А. Д.

Ю 16 Число и мысль. Вып. 8. (Математики измеряют сложность). — М.: Знание, 1985. — 192 с. (Нар. ун-т. Естественнонаучный фак.)

55 к.

40000 экз.

В книге рассматриваются проблемы и методы применения математики в планировании, управлении проектировании с позиций теории сложности.

Книга может служить пособием для слушателей народных университетов естественнонаучных знаний, а также будет полезна всем, кто интересуется современными проблемами естествознания.

Ю $\frac{0604020100-065}{073(02)-85}$ 14—85

ББК 65.9 (2) 23
33СЗ

Издательство «Знание», 1985 г.

ПРЕДИСЛОВИЕ

В послевоенные годы в различных областях науки — в автоматическом регулировании, математической экономике, биологии и ряде инженерных дисциплин — возникла проблема исследования феномена сложности. Проектирование современной техники, планирование на разных уровнях, управление системами, объяснение и прогнозирование эволюционных процессов сталкиваются с трудностями одной и той же природы. Во всех этих целенаправленных операциях требуется установить принципиальную разрешимость появляющихся здесь задач, а для алгоритмически разрешимых задач — выяснить, насколько они физически осуществимы. Другими словами, необходимо оценить минимальные объемы тех или иных ресурсов, требуемых для реализации решения. Анализ этих проблем и составляет сущность формирующейся в последние десятилетия дисциплины — теории сложности.

Теория сложности стоит на стыке философии, логики и различных разделов естественных, технических и общественных наук. В разработке категории сложности, в изучении свойств компонент сложности и в совершенствовании методов оценки составляющих сложности заинтересованы специалисты самых различных дисциплин. И с ростом масштабов производства и уровней управления, с расширением горизонтов исследований эта заинтересованность все возрастает. Трудно переоценить значение своевременного ответа на вопрос, в какой мере реально достижение той или иной цели и какие для этого необходимы ресурсы.

Советские ученые, и в первую очередь академик А. Н. Колмогоров и его ученики, первыми проложили пути к формированию категории сложности, к исследованию ее свойств и перспектив ее приложений. В последующем стало ясно, что логическое понятие алгоритмической сложности является важной, но не единственной характеристикой категории сложности. Появились и другие направления исследований, связанные с вычислительной сложностью, информационной сложностью, энергетической сложностью, сложностью схем, сложностью обработки информации

и др. Различные компоненты сложности оценивают разные аспекты трудоемкости и ресурсоемкости достижения тех или иных целей. Среди работ по различным составляющим сложности фигурируют и работы авторов настоящей книги.

Несмотря на интенсивные исследования в этой области, теория сложности еще не вступила в пору зрелости, когда ее результаты и приложения стали бы доступными для многих заинтересованных в них специалистов. Тем не менее влияние, которое могут оказать уже установленные положения теории сложности на методологию постановки и подходы к решению современных крупномасштабных задач, требует ознакомления широкого круга специалистов — экономистов и философов, инженеров и биологов — с современным состоянием теории и с перспективами ее развития и применения.

Предлагаемая вниманию читателей книга представляет собой достаточно популярное изложение этой отнюдь не элементарной проблематики, написанное специалистами для неспециалистов. Здесь охвачены различные разделы теории сложности и обсуждается достаточно широкий диапазон ее возможных приложений. Авторы, как они сами об этом заявляют, вынуждены были в угоду доступности изложения несколько поступиться формальной строгостью формулировок. Это, однако, никоим образом не повлияло на познавательную ценность авторской аргументации. Следует все же отметить, что не все приведенные в книге истолкования перспектив теории сложности могут быть безоговорочно восприняты критически настроенным читателем. Но ведь постановка острых вопросов и выявление и обсуждение правдоподобных гипотез стимулируют целеустремленное движение к истине. Уместно вспомнить замечание В. Райгли о том, что из двух искателей истины, придерживающихся одного и того же мнения, по крайней мере один лишний.

Представляется, что настоящая книга дает основание для полезных дискуссий по одной из кардинальных проблем современной науки.

Член-корреспондент
АН СССР
Я. З. Цыпкич

ЧТО ТАКОЕ КАТЕГОРИЯ «СЛОЖНОСТЬ» И ЗАЧЕМ ОНА НУЖНА?

Интуитивные определения сложности

В толковом словаре Владимира Даля (т. IV, с. 216) приведено следующее определение термина «сложность»: «Сложность — свойство и состояние «сложенного»; совокупность, весь состав, общность составного». Далее следуют пояснительные примеры. «Сложность машины затрудняет приложение к делу. По сложности дела этого следователи спутали его».

Вряд ли можно определение и пояснения Даля принять за образец формальной четкости и строгости. Однако в разговорной речи и в художественной литературе, при обсуждении повседневных дел и многих производственных вопросов использование термина «сложность», как правило, не вызывает недоразумений. В зависимости от контекста, в котором фигурирует термин «сложный», его истолковывают как «большой», «трудоемкий», «запутанный», «непредсказуемый» и т. д.

Большее внимание определению понятия «сложность» должно быть, естественно, уделено в науке, технике, экономике. Нетрудно представить себе ситуации, в которых сложность систем, явлений, процессов следует измерять и сравнивать. Тем не менее в работах, в которых подчеркивается необходимость и важность категории сложности в задачах управления, проектирования и организации, обычно не уточняется и не формализуется это понятие. Молча-

ливо предполагается, что специалисты, занимающиеся одним и тем же кругом проблем, одинаково понимают и истолковывают этот термин. Так, инженеры на эвристическом уровне подразумевают под сложной системой систему из большого числа взаимосвязанных элементов, поведение которой трудно предсказуемо. Специалисты по теории принятия решений называют сложной систему, цели поведения которой противоречивы и не всегда поддаются формализации. Математики-прикладники под сложной проблемой понимают труднорешаемую задачу, анализ которой связан с чрезмерно большим количеством вычислений. Биологи связывают понятие «сложность организма» со степенью его организации.

До последних лет отсутствие четких определений понятий «сложная система», «сложная задача», «сложный процесс» обычно не мешало их распознаванию и исследованию. Но уже сейчас можно указать математические и прикладные проблемы, для постановки и решения которых недостаточно смутных эвристических представлений. Все настоятельной ощущается необходимость в четком определении категории «сложность» и уточнении различных ее аспектов применительно к разным системам и задачам.

Подчеркнем с самого начала, что сложность систем, явлений, процессов вряд ли может быть охарактеризована единым показателем. Сложность — категория многогранная. В различных проблемах проявляются различные составляющие сложности. Системы или процессы, сложные в одних отношениях и в определенных ситуациях, могут оказаться несложными в других отношениях и в другой обстановке. Целесообразно рассматривать статическую и динамическую сложность, сложность объекта и сложность функционирования, внутреннюю сложность системы и сложность управления ею. В зависимости от класса исследуемых систем (от их назначения и условий функционирования) системы сравниваются и оцениваются по таким компонентам, как сложность описания (моделирования), алгоритмическая сложность, энергетическая сложность, информационная сложность и др. Обсуждаются и другие характеристики сложности, предназначенные для исследования и оценки биологических, экономических и социальных систем. Эти работы относятся к наиболее трудным разделам теории сложности. Они еще недостаточно формализованы и по ним среди специалистов еще не достигнуто единого мнения.

В настоящей работе кратко излагаются результаты современной теории сложности, представляющие, с нашей точки зрения, интерес для исследования закономерностей целеустремленных систем и целенаправленных процессов. Особое внимание уделяется аспектам теории сложности, важным для постановки и анализа задач принятия решений, с которыми так или иначе связаны основные проблемы планирования, управления, проектирования и организации. Естественно, что мы не можем претендовать на сколь-нибудь полное изложение основных результатов и перспектив этой новой интенсивно развивающейся дисциплины.

Оговоримся с самого начала, что по необходимости мы будем пренебрегать формальной строгостью изложения в угоду доступности и отдадим предпочтение качественному обсуждению материала по сравнению с формальным анализом.

Сложность и неформальные проблемы

Приведем неформальные высказывания некоторых видных ученых о роли понятия «сложность» в познании закономерностей реального мира.

Известные советские философы Б. В. Бирюков и В. С. Тютин пишут: «С понятием о сложности и сложных системах мы встречаемся фактически во всех — или почти во всех — науках; сложность — это общенаучное понятие, приближающееся по своему «статусу» к философской категории» [2, с. 219]. Это заключение отражает реальную динамику познавательного процесса последних десятилетий.

Один из величайших математиков XX века Джон фон Нейман говорил, что теория сложности является «предпосылкой к пониманию процессов обучения и развития» и что «понятие сложности, несмотря на его *prima facie* количественный характер, может в действительности выражать нечто качественное — иметь принципиальное значение» [32, с. 92].

Специалист по прикладной математике Дж. Касти в монографии из серии, основанной Международным институтом прикладного системного анализа в Вене, приводит следующее определение одной из важных компонент сложности. «Сложность управляемых систем, по существу, яв-

ляется мерой вычислительных возможностей, необходимых для реализации заданного поведения. В идеале математическая теория сложности должна достигнуть уровня, аналогичного уровню развития теории вероятностей. В то время как вероятность можно рассматривать как меру неопределенности в данной ситуации, сложность можно трактовать как меру понимания поведения системы» [18, с. 57—58].

Стаффорд Бир — бывший президент Британского общества по исследованию операций — называет сложность основным свойством реального мира. Он говорит: «Сложность становится проблемой века, точно так же, как умение обрабатывать природные материалы было проблемой жизни и смерти для наших праотцов» [1, с. 8]. Обилие информации, порождаемое сложностью современной жизни, представляет собой разновидность загрязнения окружающей среды. Слова-паразиты, понятия-паразиты, бесчисленные, ни о чем не говорящие данные, ложные посылки, организационные структуры, не соответствующие объективным условиям времени и места, обезоруживают человечество в борьбе со «сложностью мира». Основной способ преодоления сложности, по Бире, — создание и изменение метасистемных управляющих механизмов. Отсюда задача теории сложности — разработка емких и экономных понятий, отражающих различные аспекты категории «сложность», облегчающих обмен информацией и побуждающих к изменению образа мышления в соответствии с развитием объективной реальности.

Математические задачи теории сложности

В современной математической логике исследуются важные и трудные задачи, решение которых может иметь далеко идущие практические последствия. Это задачи, связанные с одной из наиболее интересных особенностей математического творчества — с доказательством невозможности определенных построений. Пожалуй, наиболее значительными результатами такого рода стали полученные в тридцатые годы А. Черчем, а затем в сороковые и пятидесятые годы А. А. Марковым, Э. Постом, П. С. Новиковым и в 1969 г. Ю. В. Матиясевичем доказательства алгоритмической неразрешимости некоторых массовых проблем (классов задач) математической логики и алгебры.

Позже было показано, что для некоторых достаточно общих алгоритмически неразрешимых массовых проблем отказ от решения относительно небольшой доли индивидуальных задач, входящих в массовую проблему, может сделать ее алгоритмически разрешимой. Таким образом, удалось существенно расширить класс массовых проблем, для которых доказана алгоритмическая разрешимость. Однако «чистое» доказательство алгоритмической разрешимости проблемы, т. е. доказательство существования алгоритма, позволяющего решать все индивидуальные задачи класса, еще не достаточно для того, чтобы говорить о физической возможности реализации решения.

Помимо алгоритмической неразрешимости, существует еще, если можно так выразиться, физическая неразрешимость, т. е. невозможность решения, связанная не с отсутствием разрешающего алгоритма, а с ограниченностью наших временных и материальных ресурсов. Минимальную величину времени, памяти, энергии и других ресурсов, гарантирующую решение требуемого качества любой задачи исследуемого класса (любой индивидуальной задачи рассматриваемой массовой проблемы), естественно принять за определение соответствующей компоненты сложности массовой проблемы. Аналогичные рассуждения позволяют получить умозрительные представления о составляющих сложности различных систем. Эти величины характеризуют, в частности, степень физической реализуемости постижения закономерностей состояния и поведения системы.

Пренебрежение к физической реализуемости решения алгоритмически разрешимых проблем отнюдь не способствует взаимопониманию теоретиков и практиков. Многие логически безупречные построения не воспринимаются инженерами, экономистами и физиками в силу их предельной абстрактности. Абстракция потенциальной осуществимости, которая, по А. А. Маркову [29], состоит в отвлечении от реальных границ наших конструктивных возможностей, обусловленных ограниченностью нашей жизни в пространстве и времени, приемлема для построения оснований математики, но не приемлема для анализа практических задач.

Существуют убедительные доводы в пользу того, что физические процессы или материальные объекты не могут характеризоваться такими числами, как 10^{100} и более. Реалиста-прикладника не интересует различие между чис-

лами 10^{100} и $10^{100!}$. За тем и другим числом ничего реального нет. Поэтому глубокие формальные исследования, которые связывают характеристики систем с числами подобного порядка, не способствуют сокращению разрыва между теорией и практикой.

Важная задача теории сложности — дополнить доказательства принципиальной реализуемости алгоритмов или систем оценкой их физической реализуемости — оценкой минимальных величин различных ресурсов, необходимых для этого.

Сложность и прикладные проблемы

1. Роль теории сложности в прикладных проблемах, связанных с проектированием будущего, трудно переоценить. Между желаемым и реализуемым большая дистанция. Эта дистанция измеряется, грубо говоря, компонентами сложности, представляющими собой минимальные ресурсы, необходимые для достижения цели при наиболее экономном использовании ресурсов. Можно полагать, что уже полученные результаты теории сложности имеют большое методологическое значение и должны помочь выявлять различие между реальными и ложными ориентирами.

Проблемы сложности приобрели особую актуальность в наш век, когда результаты рукотворной деятельности человека все больше влияют на будущее человечества. Миллионы лет биологической эволюции, обкатанной на миллиардах организмов, отшлифовали и упростили механизмы взаимодействия элементов живой материи. Эволюция обеспечила сложнейшие биологические системы простейшими принципами управления, содействующими развитию и совершенствованию жизни. Украинский философ XVIII века Г. С. Сковорода (1722—1792), пантеист по своим философским взглядам¹, выразил эту мысль следующим образом: «Слава тебе, Господи, что ты создал все нужное нетрудным, а все трудное — ненужным». «Трудное — нетрудное» Сковороды — это, конечно, то, что мы в этой книге называем «сложным», «менее сложным», «простым» и т. п.

¹ Пантеизм — философская концепция, отождествляющая природу и бога. Пантенстические представления нередко служили путем к материализму.

В век научно-технической революции ситуация существенно меняется. Возможности управления все больше отстают от бурного роста сложности общества, экономики, производства, техники. Слишком мало времени прошло после внедрения современных достижений науки и техники, чтобы уточнились цели (отдаленные и ближайшие) и упростились механизмы их достижения. Наука, сокращающая опыт быстротекущей жизни, играет в известном смысле роль катализатора эволюции. Новые научные понятия и категории призваны упростить восприятие сложного мира, отделить постоянно действующие факторы от проходящих, прогрессивные направления — от бесперспективных, достижимое — от нереального. Представляется, что в такого рода исследованиях ведущую роль будет играть теория сложности.

2. Приведем несколько примеров.

Интуитивно ясно, что централизованное планирование работы сложной системы обладает преимуществами перед автономным. В планировании из единого центра привлекает возможность исходить из интересов целого. При такого рода планировании можно маневрировать ресурсами и повысить, таким образом, эффективность функционирования системы. Но всегда ли реализуемы достоинства централизации, все ли ее принципиальные преимущества достижимы? Оказывается, нет. Все зависит от размерности системы. Во многих случаях оптимизация (в любом заранее заданном смысле) плана при реальном количестве факторов, подлежащем учету, — неподъемная задача при любом предвидимом состоянии вычислительной техники. Даже составление сбалансированного (не обязательно оптимального, пусть только допустимого) плана, определяемого сотнями тысяч и более номенклатур, — задача колоссальной вычислительной сложности. Практика планирования давно пришла к тому, что сохранение принципов централизации требует, по крайней мере, агрегирования элементов системы, создания иерархических структур, охватывающих относительно самостоятельные подсистемы управления по территориальному или отраслевому принципу. Однако организация процедур, упрощающих планирование и стимулирующих выполнение планов, не всегда основана на достаточно серьезной методической (логической и экономической) базе и всестороннем анализе и обобщении опыта. Нередко, к сожалению, стремление сократить трудоемкость планирования приводит к изданию

всякого рода инструкций, указаний, директив, основанных на эвристических, подчас субъективных и не всегда убедительных соображениях. В итоге эффект от принципиальных преимуществ централизованного управления ресурсами может в значительной мере смазываться.

Из зарубежной практики известны случаи, когда сложность управления заставляла крупные фирмы делиться на филиалы и допускать конкуренцию между дочерними предприятиями. Правление фирмы, объединяющее филиалы, регулировало лишь ограниченное число факторов технической и коммерческой политики.

Разумеется, конкуренция чужда нашему обществу. Отработанные в нашей стране механизмы повышения эффективности хозяйственной деятельности предполагают неантагонистическое соревнование предприятий, трудовых коллективов, рядовых тружеников и лиц, ответственных за принимаемые решения. Намеченные в ходе экономического строительства различные формы хозрасчета, представляющие определенную хозяйственную самостоятельность промышленным и сельскохозяйственным объединениям при сохранении основных рычагов управления в центре, безусловно, упрощают процедуру планирования и повышают качество выполнения планов. Выбор эффективных и реализуемых хозяйственных механизмов и отработка их структуры требуют исследования и сопоставления различных компонент их сложности. Представляется, что логические и вычислительные основы рациональной организации централизованного планирования и выбора механизмов управления хозяйством — одно из важных, хотя и весьма трудоемких приложений теории сложности.

3. В качестве второго примера рассмотрим задачу автоматизации проектирования радиоэлектронной аппаратуры. Это одна из наиболее актуальных задач современной техники, от качества решения которой кардинальным образом зависят возможности радиоэлектронной промышленности и, стало быть, уровень автоматизации ведущих отраслей производства.

Одна из типичных задач в этой области формулируется следующим образом.

На небольшой площади — на плате или на поверхности кристалла (размером меньше чем 5×5 мм) — требуется разместить десятки, сотни, а то и тысячи и десятки тысяч (а может быть, и больше) элементов и соединить их связями в соответствии с заданной схемой. В формальных

терминах это обычно задачи целочисленного программирования с огромным числом переменных и ограничений. Оценка трудоемкости решения таких задач — одна из важных проблем теории вычислительной сложности. Современное состояние теории приводит к выводу, что для схем достаточно широких классов нет и вряд ли будут разработаны методы точного или приближенного (с гарантированной точностью) решения задач размещения и трассировки за приемлемое время. Где же выход?

Один из подходов к проблеме автоматизации разработки радиоэлектронной аппаратуры заключается в организации интерактивного (диалогового — человеко-машинного) проектирования, при котором могут быть эффективно использованы вычислительная мощь машины и опыт и интуиция проектанта. Задача упрощается, если удастся, кроме того, выделить естественные содержательные признаки схем, позволяющие разделить класс схем на узкие подклассы. Чем уже класс схем, тем больше оснований ожидать создания эффективных методов проектирования, учитывающих специфику класса. Постановки обеих задач — рационального разделения функций между оператором и машиной и классификация схем — и подходы к их решению непосредственно связаны со сложностными оценками. Здесь существенную роль играют оценки сложности описания схем, сложности решения соответствующих экстремальных задач, сложности вычислений. Можно, например, ожидать, что поддаются анализу и проектированию в приемлемые сроки БИСы и СБИСы (большие интегральные схемы и сверхбольшие интегральные схемы), которые, несмотря на огромное число элементов, имеют относительно невысокую сложность описания, т. е. обладают в некотором смысле регулярной структурой. Такие схемы могут быть закодированы так, что их описание задается относительно простой программой. Практика создания БИСов и идет по этому пути. Так называемые матричные БИСы представляют собой иерархические блочные структуры, в которых повторяемость и стандартизация существенно снижают трудоемкость проектирования. Дальнейшее увеличение числа элементов в СБИСах, по-видимому, приведет к увеличению числа ступеней иерархии. Те же сложностные соображения показывают, что регулярность структуры БИСов упрощает и другие крайне трудоемкие задачи автоматизации проектирования — диагностику и контроль работоспособности сложных схем.

Представляется, что по мере усложнения радиоэлектронной аппаратуры роль различных разделов теории сложности в выявлении рациональных методов проектирования будет все возрастать.

Возможно ли путешествие по телеграфу?

В заключение вводной главы приведем пример, иллюстрирующий различие между потенциальной осуществимостью и физической реализуемостью процесса.

Мы рассчитываем на чувство юмора читателя и надеемся, что он оценит условность примера и не предъявит нам больше претензий, чем автору фабулы примера — основоположнику кибернетики Норберту Винеру.

В книге «Кибернетика и общество» Н. Винер пишет: «Мы рассматриваем два типа связи, а именно: материальные перевозки и передачу лишь одной информации, но в настоящее время человек может переезжать с одного места на другое только посредством первого типа связи, а не в качестве сигнала. Однако даже сейчас передача сигналов помогает распространять человеческие чувства и человеческие способности с одного конца света на другой». И далее: «различие между материальной транспортировкой и транспортировкой сигналов теоретически никак не является постоянным и непреходимым» [7, с. 106]. Игнорируя категорию «сложность», Н. Винер далее говорит о «путешествии по телеграфу» как об идее, «весьма близкой к истине», реализации которой мешают лишь «технические трудности».

Оценка минимальных ресурсов, необходимых для телеграфирования человека, свидетельствует, однако, что дело здесь отнюдь не только в «технических трудностях».

Приведем некоторые правдоподобные рассуждения на эту тему.

Представляются возможными два подхода к реализации идеи путешествия по телеграфу, например, через Атлантический океан. Строительный материал (элементарные частицы, атомы, молекулы), из которого составлен организм человека, имеется и по ту и по другую сторону океана. Потенциальные путешественники различаются взаимным расположением и взаимосвязями элементарных «кирпичиков» в их организме. Пересечь океан можно, в частности, если передать по телеграфу информацию о «структуре»

путешественника — информацию о взаимном расположении и о взаимосвязях «кирпичиков», составляющих его организм. Так, архитектор в Европе может по телеграфу руководить постройкой здания в Америке. «Физическая транспортировка архитектора и его документов может быть весьма эффективно заменена передачей сигналов, что не влечет за собой передвижения ни одной крупницы материи с одного конца линии на другой» [7, с. 106].

Таким образом, первый подход к реализации идеи путешествия по телеграфу связан с разворачиванием человеческого организма — с последовательным прохождением луча передатчика через все частицы, составляющие организм путешественника, — и с восстановлением организма (в соответствии с переданной по телеграфу информацией) из набора запасенных у приемника элементарных «кирпичиков».

Реализация такого подхода к путешествию по телеграфу требует преодоления по крайней мере двух препятствий. Первое из них указано самим Н. Винером: «Сохранение устойчивости организма, в то время как часть его медленно разрушается в целях пересоздания ее на другом материале где-нибудь в другом месте, повлечет за собой понижение степени деятельности организма, что в большинстве случаев разрушало бы жизнь в ткани» [7, с. 111]. Правда, Н. Винер тут же добавляет, что проблема сохранения жизни организма во время его радикальной перестройки, необходимой для телеграфной передачи строения человека, связана, по-видимому, лишь с техническими трудностями — реализуется же гораздо более радикальная перестройка организма бабочки на стадии куколки.

Второе препятствие для путешествия по телеграфу — это ограниченная пропускная способность телеграфного канала и огромный объем информации, подлежащий передаче. Г. Дж. Бремерман [3] оценивает снизу количество информации, необходимое для передачи структуры человеческого мозга. Это величина порядка 10^{12} — 10^{13} бит. Ясно, что количество информации, определяемое разверткой всего человеческого организма, а не только мозга, — существенно большая (по порядку) величина. С другой стороны, пропускная способность телеграфного канала равна $50 \div 200$ бит/с. Это значит, что объем информации, отвечающий развертке человеческого организма, столь велик, что передача его по телеграфу займет неизмеримо больше времени, чем путешествие морем или тем более по воздуху.

Больше того, это время (порядка 1000 лет) существенно превышает продолжительность человеческой жизни. Таким образом, рассуждения, оставляющие в стороне понятие «сложность», игнорируют и различие между потенциально осуществимым и физически реализуемым.

Второй подход к путешествию по телеграфу связан с передачей неизмеримо меньшего объема информации. Речь идет о передаче генетической информации — хромосомной организации индивида, определяющей его генотип. Однако таким образом через океан переправится информация, недостаточная даже для восстановления зиготы. Для физической реализации путешествия по телеграфу нужно еще на приемном пункте получить по информации о генотипе информацию об организме. По закону Геккеля филогенез повторяется в онтогенезе, т. е. каждый эмбрион повторяет до некоторой степени эволюцию, которая привела к его появлению. Генетический контроль не может охватить все связи между отдельными нейронами мозга. Большинство связей должно быть в какой-то степени случайными или построенными по некоторому фиксированному образцу. Уже указывалось, что для детального установления структуры человеческого мозга требуется как минимум 10^{12} бит информации. С другой стороны, при изучении дрозофилы установлено, что максимальная оценка информации, переносимой генами, равна 10^4 — 10^5 бит. Таким образом, для идентификации личности (для восстановления фенотипа) необходим дополнительный источник информации о связях между нейронами, установившихся в процессе развития и обучения организма. Это означает, что для восстановления фенотипа следует передать и воспроизвести информацию о среде и последовательности ситуаций, в которых развивался и обучался «путешественник». По-видимому, время, потребное для этого, существенно превышает продолжительность человеческой жизни. Путешествие по телеграфу можно связывать, таким образом, только с путешествием в будущее. К аналогичному выводу приводят и оценки других компонент сложности путешествия по телеграфу.

АЛГОРИТМИЧЕСКАЯ СЛОЖНОСТЬ

Что такое алгоритм?

1. Понятию «алгоритм» уже более тысячи лет. В течение длительного времени это понятие использовали только математики и истолковывали как последовательность правил решения задач того или иного класса или правил вычисления функции, заданной тем или иным образом. Такое интуитивное толкование алгоритма обычно устраивало пользователей. Наличие алгоритма для задач некоторого класса отождествляется с возможностью «механического» решения любой задачи класса (если, конечно, решение существует) и установления ее неразрешимости, если задача не имеет решения.

Алгоритм вычисления функции позволяет «автоматически» находить значения функции для любого набора аргументов, на котором она определена. Алгоритм для решения задач класса или вычисления значений функции избавляет пользователя от необходимости в изобретательности или творческом подходе при решении каждой конкретной задачи класса или при вычислении функции в любой точке области ее определения.

Многие великие математики прошлого верили в то, что для любого класса задач (и не только математических) можно (по крайней мере в принципе) создать алгоритм. Другими словами, предполагалось, что для каждого класса задач можно придумать автоматически работающую машину, которая в соответствии с алгоритмом решит любую задачу класса.

Почти тысячу лет не возникало необходимости в уточнении и формализации понятия «алгоритм». Среди ученых царила уверенность в том, что любая математическая задача может быть алгоритмически решена. Вера в возможность создания алгоритмических методов решения различных классов задач перекочевала из математики в другие разделы естествознания. Однако в тридцатые и последующие годы нашего века интуитивная уверенность в неограниченных возможностях алгоритмических методов была поколеблена.

В 1931 г. знаменитый австрийский математик Курт Гедель доказал алгоритмическую неразрешимость некоторой известной математической проблемы с помощью общих алгоритмов определенным образом формализованного класса. Необходимо было выяснить, совпадают ли

эти алгоритмы со всеми мыслимыми алгоритмами. А для этого следовало уточнить и формализовать интуитивное определение алгоритма. Актуальность задачи возросла в связи с появлением и развитием вычислительной техники и автоматизацией различных процессов.

Сейчас понятие «алгоритм» интересует не только формалиста-логику, но и вычислителя, составляющего программу для ЭВМ, химика, описывающего течение реакции, генетика, изучающего процесс развития организма, экономиста, планирующего хозяйственную деятельность, диспетчера, управляющего движением, лингвиста, интересующегося проблемами перевода, и т. д. Существует много путей достижения одного и того же результата, и можно многими способами вмешиваться в управляемый процесс и направлять его течение. Возникает естественное желание сформировать в каком-то смысле лучшие, наиболее экономные инструкции по организации целенаправленного процесса. Для этого необходимо перейти от интуитивного понятия алгоритма к четкому формальному.

2. После работы К. Геделя был предложен ряд уточнений понятия «алгоритм». Уточнения формулировались в различных терминах применительно к различным доступным воздействию объектам (к так называемым конструктивным объектам). Конструктивные объекты можно перенумеровать и, таким образом, всегда можно считать члены натурального ряда объектами, которыми оперирует алгоритм. Это позволяет сводить любой алгоритм к процессу детерминированных вычислений и сопоставлять между собой различные определения и уточнения понятия «алгоритм».

Все известные формализации и уточнения алгоритма, независимо от того, сформулированы ли они в терминах функций или устройств для вычисления функции, исходят из следующей модели детерминированных вычислений. Вычислительный процесс может быть представлен в виде последовательных итераций, в которых на каждом шаге используются операции из некоторого конечного набора раз навсегда заданных операций. Такой вычислительный процесс является алгоритмическим, если функции, вычисленные детерминированным итеративным образом с помощью соответствующих элементарных шагов, исчерпывают функции, допускающие вычисление любым другим вычислительным процессом, который интуитивно является алгоритмическим.

Различные конкретизации определения алгоритма в терминах машин или итеративных процессов связаны с именами А. М. Тьюринга, Э. Л. Поста, А. Черча, Дж. фон Неймана, А. А. Маркова, С. К. Клини, А. Н. Колмогорова, В. А. Успенского, Р. Петер. Все известные формальные определения алгоритма эквивалентны между собой. В настоящее время существует убеждение, что все упомянутые определения алгоритма адекватно отражают смысл, интуитивно вкладываемый в термин «алгоритм». Это так называемый тезис Черча, который отождествляет интуитивное понятие «алгоритм» с любым из эквивалентных между собой точных определений.

Тезис Черча представляет собой естественнонаучную гипотезу. Он не может быть формально доказан, поскольку не существует точного определения понятия «алгоритм» в интуитивном смысле. Однако тезис Черча выдержал все проверки на соответствие интуитивным представлениям об алгоритме. Можно сказать, что тезис Черча — это экспериментальный факт, подтверждающийся по крайней мере следующими свидетельствами:

а) все известные в теории и на практике алгоритмы могут быть «промоделированы» любым из алгоритмов в точном определении;

б) указаны подходящие нумерации входов и выходов схем переработки информации, соответствующих известным определениям алгоритмов, при которых все схемы обработки информации приводят к одному и тому же результату;

в) в литературе имеется огромный набор описаний различных вычислительных процессов в терминах различных определений алгоритма, в частности в терминах частично-рекурсивных функций.

Ниже будут приведены две формализации понятия «алгоритм» — в терминах функций (частично-рекурсивных функций) и в терминах вычислительного устройства (машины Тьюринга).

Уточнение и конкретизация понятия «алгоритм» позволят далее четко поставить вопрос о сложности алгоритма.

Частично-рекурсивные функции

1. Опишем вычислительный процесс, который может быть принят в качестве формального определения алгоритма. Процесс итеративным образом использует набор из трех базовых функций и трех элементарных операций.

Введем предварительно несколько определений.

Пусть X и Y два произвольных множества. Отображение $f: X \rightarrow Y$ называется **частичной функцией**, если область определения f $Df \subseteq X$. Понятие частичной функции вводится для того, чтобы подчеркнуть, что далеко не всякий осмысленный вычислительный процесс приводит в соответствие каждому значению аргумента $x \in X$ значение функции $f(x) \in Y$. На множестве $X \setminus Df$, если оно не пусто, значение f не определено.

В дальнейшем мы будем рассматривать частичные функции из $X = Z_+^m$ в $Y = Z_+^n$, где $Z_+ = \{1, 2, \dots\}$ — множество натуральных чисел, а Z_+^m — m -кратное декартово произведение Z_+ на само себя — множество упорядоченных наборов целых положительных чисел (x_1, \dots, x_m) , $x_i \in Z_+$, $i = 1, \dots, m$. В теории алгоритмов члены натурального ряда часто представляют в двоичном коде — словами в алфавите $\{0, 1\}$, т. е. конечными последовательностями из нулей и единиц. Иногда нам будет удобно обращаться к двоичному кодированию входа и выхода вычислительного процесса.

Дальше мы будем вынуждены несколько поступиться строгостью изложения и пользоваться термином «программа», понимая под ним программу для универсальной вычислительной машины, построенную без учета ограничений на время счета и потребную память.

Будем называть частичную функцию f из Z_+^m в Z_+^n **вычислимой**, если существует программа, которая за конечное время сопоставляет входу $x \in Z_+^m$ выход $f(x) \in Z_+^n$ в случае, когда $x \in Df$, и не останавливается (или останавливается при $y \notin Z_+^n$) для всякого $x \in Z_+^m \setminus Df$. В противном случае частичная функция f называется **невычислимой**.

В качестве частичных функций можно, в частности, рассматривать классы задач. Значения аргументов такой функции — условия индивидуальных задач класса. Соответствующие значения функции — решения этих задач.

Вычислимые функции, как видно из их определения, как раз и являются теми функциями, для которых могут быть построены вычислительные процессы (программы, алгоритмы), позволяющие «автоматически» за конечное время установить значение функции в любой точке, в которой она определена, или сообщить о том, что это невозможно сделать.

На первый взгляд может показаться, что все мыслимые частичные функции вычислимы. Действительно, постро-

ить и описать невычислимую функцию отнюдь не просто. Способы их задания непривычны для неспециалиста. Тем не менее невычислимые функции существуют. Каждая программа — это конечный текст в конечном алфавите. Следовательно, множество вычислимых функций, как и множество программ, счетно, в то время как множество всех функций из Z_+ в Z_+ несчетно. Это нестрогая аргументация. Ее уязвимость подчеркнута в [28]. Тем не менее высказанное утверждение верно. В [28] приведены примеры невычислимых функций. Алгоритмическая сложность $K(y|x)$ — предмет изучения настоящей главы — невычислимая функция x .

2. Приведем конструктивное описание вычислимых функций.

Оказывается, можно указать всего три простейшие (базовые) функции и три элементарные операции над ними, с помощью которых удастся итеративным путем получить любую вычислимую функцию. Базовые функции и элементарные операции над ними и используются для уточнения и формализации понятия «алгоритм».

Назовем базовыми функциями следующие три функции:

а) функция следования $C^1(x): Z_+^1 \rightarrow Z_+^1$, $C^1(x) = x + 1$ — функция одного целочисленного переменного, определяющая натуральное число, непосредственно следующее за значением аргумента;

б) тождественная (нулевая) функция $0^n(x_1, \dots, x_n): Z_+^n \rightarrow Z_+^1$, $0^n(x_1, \dots, x_n) = 0$ — функция n независимых переменных, тождественно равная нулю;

в) функция проектирования $Pr_i^n(x_1, \dots, x_n): Z_+^n \rightarrow Z_+^1$, $Pr_i^n(x_1, \dots, x_n) = x_i$ — функция n независимых переменных, значение которой при любом наборе аргументов равно значению i -го переменного.

Введем три элементарные операции над базовыми функциями и результатами оперирования с ними.

1) Композиция (суперпозиция или подстановка) h . Оператор композиции ставит в соответствие паре частичных функций $f: Z_+^m \rightarrow Z_+^n$ и $g: Z_+^n \rightarrow Z_+^p$ частичную функцию $h = g \circ f: Z_+^m \rightarrow Z_+^p$, $h(x) = (g \circ f)x = g(f(x))$.

2) Рекурсия. Операция рекурсии сопоставляет паре частичных функций $f: Z_+^n \rightarrow Z_+^1$ и $g: Z_+^{n+2} \rightarrow Z_+^1$ частичную функцию $h = Z_+^{n+1} \rightarrow Z_+^1$, определяемую рекурсией по последнему аргументу $h(x_1, \dots, x_n, y + 1) = g(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y))$, $h(x_1, \dots, x_n, 0) = f(x_1, \dots, x_n)$.

3) Минимизация. Оператор минимизации ставит в соот-

вместо частичной функции $f: Z_+^{n+1} \rightarrow Z_+^1$ частичную функцию $h: Z_+^n \rightarrow Z_+^1$,

$$h(x_1, \dots, x_n) = \min' \{x_{n+1} \mid f(x_1, \dots, x_n, x_{n+1}) = 1\}.$$

Уравнение $f(x_1, \dots, x_n, x_{n+1}) = 1$ вводится для обеспечения однозначности $h(x_1, \dots, x_n)$.

Введенная операция \min' отличается от обычной операции нахождения минимального неотрицательного целочисленного корня уравнения $f(x_1, \dots, x_n, x_{n+1}) = 1$ тем, что решение вычисляется последовательным перебором значений $x_{n+1} = 1, 2, \dots$, пока уравнение не удовлетворится. Результат считается неопределенным, если, не дойдя до решения, мы наталкиваемся на неопределенное значение f либо, если решения не существует. Так, например, $\min \{x \mid x - 5 = 1\} = 6$, но $h = \min' \{x \mid x - 5 = 1\}$ не определено, так как при $x = 1$ значение h отрицательно.

Ясно, что операции суперпозиции и рекурсии, если их применять к всюду определенным функциям, дадут всюду определенные функции.

В отличие от них минимизация может приводить к не полностью определенным функциям. Именно этой операции мы обязаны возникновением из всюду определенных функций частичных функций.

Теперь мы подготовлены к тому, чтобы ввести важное для настоящей главы понятие рекурсивных функций.

Функция f называется **частично-рекурсивной** функцией, если она может быть получена из базовых функций $C^1(x)$, $O^n(x_1, \dots, x_n)$, $Pr_i^n(x_1, \dots, x_n)$ с помощью конечного числа операций композиции, рекурсии и минимизации.

Всюду определенная частично-рекурсивная функция называется **общерекурсивной**.

Частично-рекурсивные функции, построенные без применения оператора минимизации, называются **примитивно рекурсивными**. Ясно, что они являются и общерекурсивными.

3. Определение частично-рекурсивных функций удобно для последующего обсуждения представить в таком виде.

Последовательность частичных функций f_1, \dots, f_N называется **частично-рекурсивным описанием** функции $f = f_N$, если f_1 — одна из базовых функций, а f_i , $i \geq 2$ является либо базовой функцией, либо может быть получена применением одной из элементарных операций к некоторым из функций f_1, \dots, f_{i-1} . Функция f называется **частично-рекурсивной**, если она допускает частично-рекурсивное описание.

Функции, используемые в математике или в приложениях, обычно являются частично-рекурсивными либо (в непрерывном случае) могут быть сколь угодно точно приближены такими функциями.

А. Черч высказал предположение, что понятием частично-рекурсивной функции и исчерпывается понятие вычислимой функции. Описание частично-рекурсивной функции индуцирует понятие алгоритма, вычисляющего значения функции в области ее определения. В терминах частично-рекурсивных функций тезис Черча формулируется следующим образом.

Каков бы ни был алгоритм, определяющий значения вычислимой функции, существует эквивалентный ему алгоритм, представляющий собой описание частично-рекурсивной функции.

Машина Тьюринга

1. Приведем теперь другое уточнение понятия «алгоритм» — на этот раз в терминах некоторой идеальной вычислительной машины с неограниченными ресурсами.

Идея уточнения и формализации понятия «алгоритм», использующая воображаемую машину, принадлежит английскому математику А. М. Тьюрингу. Она была предложена в 1937 г. еще до появления электронных вычислительных машин.

«Конструкция» машины Тьюринга определяет некоторый вычислительный процесс, который и предлагается принять в качестве уточнения понятия «алгоритм».

2. Опишем «конструкцию» и принцип работы машины Тьюринга.

Машина состоит из четырех частей: ленты, считывающей головки, внутренней памяти и управляющего механизма. Лента представляет собой неограниченную в обе стороны полосу, разбитую на бесчисленное множество ячеек, нумеруемых целыми числами. (Неограниченность ресурсов — важная особенность машины Тьюринга, позволяющая доказать эквивалентность вычислительного процесса, определяемого ею, другим формализациям понятия «алгоритм». Машина Тьюринга, таким образом, не реальная, а воображаемая машина, используемая не для вычислений, а для теоретического анализа.) Лента предполагается направленной: слева от ячейки j расположена ячейка $j-1$, а

справа — ячейка $j+1$. В каждой ячейке может быть записан один из трех символов — 0, 1 или особый символ \emptyset . В начальный момент во всех ячейках, кроме конечного их числа, указаны символы \emptyset . В остальных ячейках, расположенных подряд, записаны символы 0 или 1, образующие перерабатываемое слово.

Считывающая головка может перемещаться вдоль ленты так, что в каждый момент времени она находится против одной определенной ячейки и может «обозревать» ее содержимое. В начальный момент головка находится против самой левой из ячеек ленты, содержащих символы перерабатываемого слова.

Внутренняя память — это устройство, которое в каждый момент может находиться в одном из конечного числа k состояний q_1, \dots, q_k . Среди этих состояний выделены два — начальное и так называемое стоп-состояние. В начальный момент внутренняя память находится в начальном состоянии.

Управляющий механизм — это устройство, которое в зависимости от символа, обозреваемого считывающей головкой, и состояния внутренней памяти может изменить символ, записанный в обозреваемой головкой ячейке, изменить состояние внутренней памяти и сдвинуть считывающую головку на одну ячейку вправо или влево (или оставить на месте).

Работа машины определяется ее программой — набором инструкций вида $q_i S_j \rightarrow S_i \pi q_r$, где $1 \leq i, r \leq k$, S_j, S_i принимают значения 0, 1, \emptyset , а π — один из трех символов П, Л или М, означающих указание сдвинуться на ячейку вправо, влево или остаться на месте. При этом в программе имеется одна и только одна инструкция с левой частью $q_i S_j$ для каждого возможного значения пары q_i, S_j , если только q_i — не стоп-состояние.

3. Машина работает в соответствии со следующими правилами. На каждом $(t$ -м) такте, начиная с начального, определяется состояние $q_{i(t)}$ внутренней памяти машины и считывается содержимое $S_{j(t)}$ обозреваемой ячейки. Если $q_{i(t)}$ есть стоп-состояние, то работа машины прекращается. Если состояние $q_{i(t)}$ не есть стоп-состояние, то в программе отыскивается инструкция с левой частью $q_{i(t)} S_{j(t)}$. Пусть соответствующая правая часть $S_i \pi q_r$. Тогда:

а) в обозреваемую ячейку записывается символ S_i вместо $S_{j(t)}$;

б) в качестве состояния внутренней памяти в момент $t+1$ устанавливается q_r ;

в) считывающая головка остается на месте, если $\pi=M$, сдвигается на ячейку вправо, если $\pi=P$, либо на ячейку влево, если $\pi=L$.

На этом такт работы машины заканчивается.

Результат работы после остановки машины считается определенным, если на ленте нет символов подмножества $\{0, 1\}$, разделенных символами \emptyset (т. е. либо на ленте вообще нет символов 0 или 1, либо имеющиеся символы из множества $\{0, 1\}$ идут подряд на некотором числе ячеек ленты). Если на ленте не содержится символов 0 или 1, то результат работы машины на исходном слове не определен. Если на ленте символы 0 и 1 идут подряд на некотором числе ячеек, то соответствующее двоичное слово есть результат работы машины.

4. Функционирование машины Тьюринга индуцирует вычислительный процесс, используемый для уточнения понятия «алгоритм». Алгоритм, реализуемый на машине Тьюринга, определяется парой, составленной из алфавита $Q = (q_1, \dots, q_k)$ состояний внутренней памяти и программы. Каждый такт алгоритма, примененный к записи на ленте двоичного слова x , приводит к одному из трех возможных результатов:

а) переводит записанное на ленте двоичное слово x в двоичное слово-результат;

б) приводит к остановке с неопределенным результатом;

в) останавливается на данном x

В случае, когда результат на входе x определен, число тактов работы машины на слове x до остановки называется временем работы соответствующего алгоритма на слове x .

5. Мы описали простейшую модель машины Тьюринга. Рассматриваются и различные усложнения машины, позволяющие, в частности, сократить время работы соответствующих алгоритмов на наборах входных слов. Алгоритмы, реализуемые на этих машинах, определяются парой — «структурой» и «программой». «Структура» характеризует конструктивные особенности и параметры машины, влияющие на вычислительный процесс. «Программа» — это набор инструкций, в соответствии с которыми функционирует машина.

Уточнение понятия «алгоритм» может быть проведено и на более простой машине, чем машина Тьюринга, например, на машине Поста. Ее элементарные действия проще, чем

элементарные операции машины Тьюринга, и способы записи информации менее разнообразны. Поэтому запись и переработка информации на машине Поста требуют, как правило, большего объема памяти и большего числа шагов (большего времени работы), чем описанная здесь простейшая модель Тьюринга.

Тезис Черча в терминах машины Тьюринга формулируется следующим образом.

Каков бы ни был алгоритм, определяющий значения вычислимой функции, существует эквивалентный ему алгоритм, реализуемый на машине Тьюринга.

Интуитивно ясно, что понятие «алгоритм» может быть уточнено не только в терминах простейших идеализированных вычислительных устройств, таких, как машина Тьюринга или машина Поста. Любая универсальная вычислительная машина произвольной структуры, располагающая, однако, неограниченными ресурсами, может быть использована для формализации понятия «алгоритм». Алгоритм, реализуемый машиной, определяется парой: «структурой» машины и «программой». Конечно, строгая формулировка этого утверждения (особенно четкое определение понятий «структура» и «программа») требует уточнения. Мы, однако, апеллируем здесь к интуиции читателя и ограничимся следующим формальным замечанием.

Под «структурой» вычислительной машины мы будем в дальнейшем подразумевать функцию $\varphi(p, x) = y$, ставящую в соответствие входу x и «программе» p выход y .

Тезис Черча во введенных терминах можно сформулировать следующим образом. «Структура» универсальной вычислительной машины, определяющей вместе с «программой» алгоритм, является частично-рекурсивной функцией.

Определение алгоритмической сложности

1. Как мы видели, значения вычислимой функции могут быть получены с помощью разнообразных итеративных процессов, составленных из базовых функций и элементарных операций, или с помощью алгоритмов, реализованных на машинах Поста или на разных моделях машины Тьюринга. Наконец, значения вычислимой функции могут быть получены с помощью программы для произвольной универсальной машины с неограниченными ресурсами (времени и

памяти). В любой из этих моделей существует бесконечное число «программ», позволяющих по заданному значению аргумента вычислимой функции установить или вычислить решение y задачи x . Естественно после уточнения и формализации понятия «алгоритм» сделать следующий шаг — научиться сравнивать трудоемкость решения задач на различных «структурах».

По-видимому, одной из важных характеристик сложности решения задачи является кратчайшая длина программы, содержащей всю информацию, необходимую для обеспечения обусловленного алгоритмом вычислительного процесса при помощи универсальной машины фиксированной структуры.

Пусть $x \in X$ — некоторая задача из класса X задач, y — ее решение — элемент некоторого множества Y . Уже отмечалось, что соответствие между решениями задач класса и условиями задач удобно рассматривать как частичную функцию. Алгоритм решения задачи класса X позволяет с помощью программы p получить на универсальной машине со структурой $\phi(p, x)$ решение y задачи $x \in X$.

Заметим, что не всегда удобно интерпретировать элементы x и y как задачу и ее решение. В таких случаях будем считать x и y последовательностями в некотором алфавите, истолковываемыми в соответствии с рассматриваемой ситуацией. Вместо алгоритма решения задачи будем тогда говорить об алгоритме, позволяющем с помощью программы p восстанавливать по последовательности $x \in X$ последовательность $y \in Y$ (или по коду объекта x код объекта y).

Пусть $l(p)$ — длина программы p — число знаков в ее записи. Будем считать далее, что программа записывается в двоичном коде и представляет собой, следовательно, двоичную последовательность (двоичное слово) конечной длины. Обозначим через S множество всех конечных последовательностей из нулей и единиц, т. е. множество всех допустимых программ.

Естественно под сложностью вычисления решения y задачи x (или под сложностью восстановления последовательности y по последовательности x) на машине со структурой $\phi(p, x)$ понимать минимальную длину допустимой программы p , т. е. программы, для которой $\phi(p, x) = y$. Сложность задачи бесконечна, если не существует программы, обеспечивающей вычисление решения y задачи x на машине заданной «структуры» ϕ .

Формально определение сложности вычисления реше-

ния y задачи x (или условной сложности объекта y относительно объекта x) записывается в виде

$$K_{\varphi}(y|x) = \begin{cases} \min l(p) : \varphi(p, x) = y, \\ \infty : \forall p \in S \quad \varphi(p, x) \neq y. \end{cases}$$

Величина $K_{\varphi}(y|x)$ называется также **алгоритмической сложностью**. Это название обусловлено алгоритмическим подходом к определению понятия «сложность» и отнюдь не связано с описанием сложности алгоритма решения массовой задачи. Минимальная длина программы для решения разных задач класса X может зависеть от индивидуальной задачи $x \in X$.

2. Как видим, сложность объектов (последовательностей) зависит от «структуры» вычислительной машины, на которой эти объекты реализуются. Основным результатом теории алгоритмической сложности, принадлежащий А. Н. Колмогорову и Р. Соломонову, показывает, что эта зависимость не очень существенна. А. Н. Колмогоров доказал следующее утверждение [16].

Существуют универсальные вычислительные машины со структурой ψ , такие, что соответствующие им сложности $K_{\psi}(y|x)$ переработки x в y связаны с алгоритмической сложностью $K_{\varphi}(y|x)$ решения тех же задач на машинах с любой другой структурой φ неравенством

$$K_{\psi}(y|x) \leq K_{\varphi}(y|x) + C_{\psi\varphi},$$

где $C_{\psi\varphi}$ — некоторая константа, зависящая лишь от структуры ψ и φ и не зависящая от x и y .

Это значит, что существуют универсальные машины (будем их называть оптимальными), позволяющие решать **любые** задачи с длиной программы, которая превосходит длину программы на любой другой машине не более чем на константу, зависящую лишь от этой второй машины. Определение алгоритмической сложности в соответствии с теоремой Колмогорова становится инвариантным относительно вычислительной машины, на которой реализован алгоритм. Идея доказательства состоит в том, что всякая машина «структуры» φ может быть промоделирована на универсальной машине «структуры» ψ с помощью некоторой программы (транслятора). Если y вычисляется по x в машине φ с помощью программы p , то на машине ψ он может быть вычислен программой, полученной из p при-

соединением транслятора, размер которого не зависит от x и y . Строгое обоснование теоремы Колмогорова, впрочем, как и ряда других утверждений теории алгоритмов, опирается на введенное в формальной логике понятие нумерации.

Входы и выходы алгоритмов и программы вычислений представляют собой конечные тексты — конечные последовательности букв (двоичных знаков) и как таковые они могут быть перенумерованы. Манипулирование нумерациями конечных объектов является одним из важных средств установления взаимосвязей между характеристиками объектов в теории алгоритмов. В частности, ряд результатов теории алгоритмов основан на следующем утверждении.

Пусть $v = \varphi(u)$ — функция от $u \in X$ со значениями $v \in Y$, $n(u)$ и $n(v)$ — номера u и v в некоторой нумерации. Функция φ частично-рекурсивна тогда и только тогда, когда она порождается частично-рекурсивным преобразованием Φ номеров $n(v) = \Phi[n(u)]$.

«Структура» вычислительной машины — частично-рекурсивная функция $\varphi(p, x) = y$, связывающая вход x , выход y и программу p . Чтобы вычислить выход y (решение задачи) по входу x (по условиям задачи) на машине заданной «структуры», нужно задать конечную информацию — программу p . Приведем в соответствие заданной «структуре» машины некоторую нумерацию и перенесем ее на все изучаемые объекты.

Теорема Колмогорова утверждает, что существуют (оптимальные) «структуры» ψ вычислительных машин, которым соответствуют нумерации, позволяющие наиболее экономным путем восстановить решение y задачи по ее условиям x и программе p , заданных в терминах «структуры» φ .

Именно для вычисления y по x на универсальной машине со «структурой» ψ достаточно не больше

$$K_{\psi}(y|x) + \text{const}$$

бит информации: к программе p нужно добавить номер «структуры» φ в нумерации, отвечающей «структуре» ψ (не зависящей от x и y).

Сложности $K_{\psi_1}(y|x)$ и $K_{\psi_2}(y|x)$ алгоритма, перерабатывающего x в y , относительно оптимальных «структур» ψ_1 и ψ_2 различаются не более чем на константу C_{φ, ψ_i} . Определяя алгоритмическую сложность, мы будем фиксировать

раз и навсегда некоторую оптимальную (или, как ее еще иногда называют, универсальную) «структуру».

Неопределенность константы в определении алгоритмической сложности оставляет, конечно, неприятный осадок. Можно было бы избежать неопределенности, ограничив анализ фиксированными «структурами». Вряд ли это, однако, возможно без явного произвола. А. Н. Колмогоров считает, что различные «разумные» варианты будут приводить к оценкам сложностей, различающимся на сотни бит, а не на десятки тысяч бит. И сложность «сложных» объектов, оцениваемых десятками и сотнями тысяч бит (только для столь сложных объектов и имеет смысл вся излагаемая здесь теория), будет оцениваться практически однозначно.

Это, конечно, не формально доказываемое утверждение, а интуитивные представления крупнейшего ученого — основателя теории алгоритмической сложности и не только ее.

Свойства алгоритмической сложности

1. Приведенное формальное определение сложности провоцирует вопрос: а нельзя ли построить алгоритм, вычисляющий сложность алгоритма, предназначенного для решения задач того или иного класса (для определения значений той или иной вычислимой функции). Такой алгоритм позволял бы определять длины кратчайших программ решения задач. А это шаг к автоматизации «оптимального» программирования.

Оказывается, однако, что такого алгоритма нет (идея доказательства этого факта приведена ниже при обсуждении сложности объекта). Алгоритмическая сложность является невычислимой функцией и, следовательно, разработка программ кратчайшей длины не может быть автоматизирована. Вычисление алгоритмической сложности и построение кратчайшей программы представляют собой творческие процессы, требующие в каждом отдельном случае изобретательности.

Тем не менее можно найти мажоранты алгоритмической сложности — оценки сложности сверху — такие частично-рекурсивные (и даже общерекурсивные) функции $\kappa(y|x)$, что $K_{\Phi}(y|x) \leq \kappa(y|x) + C$, где C — константа, не зависящая от x . Более того, если известна задача x и некоторое число $s \geq K_{\Phi}(y|x)$, то можно эффективно найти одну из

программ p решения задачи x на машине со «структурой» Φ , которая хотя и не будет самой короткой, все же будет иметь длину, не превышающую s .

Имеет место следующее утверждение.

Пусть объект x принадлежит конечному множеству из M объектов. Тогда доля тех объектов y , для которых условная сложность $K_{\Phi}(y|x)$ не превышает $l(M) - m$, не превосходит $\frac{1}{2}^{m-1}$. (Здесь $l(M)$ — длина двоичного представления M .)

Этот результат получается из мощностных соображений, впервые использованных К. Шенноном. Мощностные соображения позволяют получать нижние оценки сложности объектов из заданных классов при различных определениях сложности. Число задач y из заданного множества, имеющих при фиксированных x сложность не больше $l(M) - m$, не превосходит числа $2^{l(M)-m} \leq M/2^{m-1}$ всех двоичных программ размера не выше $l(M) - m$ и составляет $\frac{1}{2}^{m-1}$ часть из M объектов класса.

Особый интерес (и содержательный, и вычислительный) представляет мажоранта алгоритмической сложности $K_{\Phi}(y|x)$, являющаяся «подправленным количеством информации, содержащимся в x об y ». Подробнее об этом несколько далее.

2. Перспективность введенного А. Н. Колмогоровым понятия алгоритмической сложности и роль этой категории в теоретической математике, в частности в математической логике, не вызывает сомнений у специалистов. Не все, однако, верят в прикладную ценность теории алгоритмической сложности. Мы разделяем мнение тех, кто считает, что категория сложности безусловно перспективна и с практической точки зрения. Но мы не можем согласиться с теми, кто считает, что интерес к ожидаемым приложениям алгоритмической сложности в значительной мере снижается из-за того, что $K_{\Phi}(y|x)$, измеряя длину программы решения задачи, не касается времени, в течение которого эта программа должна работать, не оценивает требуемую память и не приспособлена к учету возможного сокращения времени решения задачи за счет распараллеливания вычислений, удлинения программы и других рационализаторских предложений.

Умозрительные рассуждения приводят к выводу, что в теории сложных систем оптимизация статических характеристик вряд ли меньше влияет на качество функционирования систем, чем оптимизация ее динамических характеристик. Во многих случаях, если не всегда, при проек-

тировании сложных систем оптимизация (в определенном смысле) внутреннего механизма действия системы является гораздо более естественным требованием к проекту, чем оптимизация реакции на выбор заданных возмущений системы.

Природа, совершенствуя свое творение — живой мир, заботилась, по-видимому, не столько о том, чтобы минимизировать время реакции живых организмов на ожидаемые возмущения внешней среды, сколько об эффективности — качестве и компактности механизма развития и адаптации — программы развертывания жизни. Эволюция живого — это последовательное совершенствование генетической программы и алгоритмов обучения. Таким образом, природа оптимизирует статические характеристики живых организмов (их алгоритмическую сложность) и не всегда стремится к оптимизации динамических характеристик организмов — времени переработки информации, поступающей от внешней среды, выбора и реализации решений. Все эти этапы — существенные элементы жизненного процесса и в ряде случаев стремление сократить их продолжительность — равносильно стремлению сократить удовлетворение, доставляемое жизнью.

Аналогичным образом рациональная экономическая система — это прежде всего производственные мощности и компактный внутренний механизм управления — программа бесперебойного функционирования экономики. Раздутые штаты, многоступенчатое управление, бесконечные инструкции, регламентирующие каждый шаг производства, — чрезмерная плата за оптимизацию реакции системы на каждое ожидаемое возмущение. Более естественный путь обеспечения рационального функционирования экономики — развитие минимального числа внутренних стимулов, определяющих программу алгоритма реакции на спрос и предложение.

Не в анализе ли сложных систем, подобных тем, которые изучаются в генетике и экономике, — ближайшие перспективы приложений теории алгоритмической сложности?

Колмогоровские результаты в теории сложности — формально сформулированные законы природы. Их интерпретация в терминах конкретных сложных систем — важная прикладная задача.

1. Введенное выше понятие алгоритмической сложности $K_\Phi(y|x)$ — сложности переработки слова x в слово y на машине со «структурой» Φ — порождает более простое понятие сложность $K_\Phi(y)$ объекта y — сложность слова y по отношению к машине со «структурой» Φ .

Будем называть сложностью $K_\Phi(y)$ объекта y значение $K_\Phi(y|x)$ при фиксированном значении x (например, при $x=1$). Таким образом,

$$K_\Phi(y) = \begin{cases} \min l(p) : \Phi(p, 1) = y, \\ \infty : \forall p \in S \quad \Phi(p, 1) \neq y. \end{cases}$$

Как и в алгоритмической сложности $K_\Phi(y|x)$, для сложности $K_\Phi(y)$ объекта могут быть найдены так называемые оптимальные структуры ψ , такие, что

$$K_\psi(y) \leq K_\Phi(y) + C_{\psi\Phi},$$

где константа $C_{\psi\Phi}$ зависит только от структур ψ и Φ и не зависит от объекта y . Это значит, что существуют такие «структуры» универсальных вычислительных машин с неограниченными ресурсами (такие частично-рекурсивные функции), в терминах которых могут быть построены программы, восстанавливающие (описывающие) любые объекты y , грубо говоря, наиболее экономным путем.

Сложность $K_\Phi(y)$ объекта y (слова y) удовлетворяет следующим свойствам:

1) $K_\Phi(y) \leq l(y)$ (следовательно, $K_\Phi(y) < \infty \quad \forall y \in S$);
 2) доля слов y , для которых $K_\Phi(y) < l_0 - m$, среди всех слов y длиной l_0 не превосходит $1/2^{m-1}$ (т. е. оценка $K_\Phi(y) \leq l(y)$ для большинства слов точная);

3) $\lim_{y \rightarrow \infty} K_\Phi(y) = \infty$. Следовательно, и $\lim_{y \rightarrow \infty} m(y) = \infty$, где $m(y) = \min_{x \geq y} K_\Phi(x)$ — наибольшая монотонно неубывающая

функция, ограничивающая $K_\Phi(y)$ снизу. При этом $m(y)$ растет медленнее любой монотонной общерекурсивной функции. Это означает, что функция $K_\Phi(y)$ не допускает нетривиальной (отличной от константы) оценки снизу;

4) $K_\Phi(y)$, как и $K_\Phi(y|x)$ — невычислимая функция. Действительно, если бы функция $K_\Phi(y)$ была вычислимой, то путем последовательного просмотра объектов (слов) y

в «естественном» порядке можно было бы найти первый из них, для которого $K_{\Phi}(y) \geq k$ ($k < l(y)$ — некоторое число). Этот объект однозначно задается числом k (программой длины $l(k)$) и программой вычисления функции K_{Φ} . Поэтому $K_{\Phi}(y) \leq l(k) + C \leq \log_2 k + C + 1$, что при достаточно больших k противоречит неравенству $K_{\Phi}(y) \geq k$. Итак, не существует алгоритма вычисления $K_{\Phi}(y)$, хотя имеется возможность получить оценки $K_{\Phi}(y)$ сверху.

Общность определений и результатов теории алгоритмической сложности (и сложности объектов) получена ценой некоторых практически нереализуемых предположений. Дело в том, что, говоря о вычислительных машинах, реализующих алгоритм, мы не ограничивали времени счета и необходимой памяти. В случае, когда на время работы (число шагов) алгоритма наложены даже самые слабые ограничения, длина программы может существенно возрасти. В литературе имеются многочисленные примеры, подтверждающие этот вывод. Качественно объяснить подобное явление можно, если учесть, что сложность — невычислимая функция. Оптимальный алгоритм будет перебирать все возможные программы, вычисляющие решение задачи или восстанавливающие объект на универсальных вычислительных машинах, и нельзя заранее сказать, какая именно программа будет самой короткой. Поэтому, если время работы машины заранее ограничено, решение задачи может быть не достигнуто или объект может быть точно не восстановлен к моменту прекращения счета.

2. Хотя, как уже указывалось, не существует алгоритмов вычисления $K_{\Phi}(y)$ и $K_{\Phi}(y|x)$ для конкретных задач или объектов, все же можно получить почти всюду сколь угодно хорошие оценки сверху этих величин.

Для приложений представляют интерес мажоранты сложности, связанные с условной или безусловной энтропией объекта.

Пусть проводится m испытаний с s возможными исходами, причем частость k -го исхода, $k=1, \dots, s$, равна p_k . Предположим, что каждый исход записывается двоичным словом длиной r . Всего таких слов 2^r . Следовательно, $s \leq 2^r$. Отождествим формальную запись рассматриваемого объекта y с двоичным словом длиной mr , представляющим собой запись исходов всей серии испытаний.

В приведенных понятиях может быть доказано следующее утверждение, устанавливающее вычислимую мажоранту сложности. Существуют константы C_{Φ} и C'_{Φ} , не завися-

щие от y , такие, что

$$K_{\Phi}(y) \leq m [H(p) + \alpha(m)] + C_{\Phi},$$

где

$$H(p) = - \sum_{k=1}^{2^r} p_k \log_2 p_k,$$

$$\alpha(m) = C_{\Phi} \frac{\log_2 m}{m} \rightarrow 0 \text{ при } m \rightarrow \infty.$$

Это значит, что энтропия Шеннона $H(p)$ может быть использована для построения мажоранты сложности $K_{\Phi}(y)$ объекта y . Энтропия тем лучше оценивает сложность объекта, чем больше число m наблюдений за его состоянием.

Соотношение между энтропией и сложностью установлено для объектов (текстов), представляющих собой запись последовательности независимых испытаний, т. е. для текстов, не имеющих других закономерностей, кроме частотных. В этом случае появляются веские основания рассматривать энтропию объекта как некоторую меру сложности объекта и, наоборот, сложность объекта как некоторую характеристику его неопределенности (хаотичности). Интуитивно ясно и далее это будет аргументировано, что, чем больше закономерностей в тексте, тем меньше его сложность. Таким образом, учет одних лишь частотных закономерностей текста и пренебрежение всеми другими (возможно, скрытыми) закономерностями приводит к завышению алгоритмической сложности текста. Это служит неформальным объяснением целесообразности использования относительно легко вычисляемой энтропии Шеннона в качестве мажоранты невычислимой алгоритмической сложности.

Сложность и случайность

1. Интуитивно ясно, что «сложные последовательности ведут себя как случайные», а «случайность» есть «отсутствие закономерности». А. М. Колмогорову и П. Мартин-Лефу принадлежит заслуга установления четкой связи между сложностью и случайностью, имеющей далеко идущие философские, математические и прикладные последствия.

Проблема обоснования теории вероятностей и ее основ-

ного понятия «случайность» (ниже «случайная последовательность нулей и единиц») имеет более чем 60-летнюю историю.

Первое определение «случайной 0—1 последовательности» принадлежит Р. Мизесу. В 1919 г. Р. Мизес предложил называть бесконечную последовательность $x = \{x_1, x_2, \dots\}$ из нулей и единиц **случайной**, если она удовлетворяет двум свойствам:

а) относительная частота появления 1 равна $1/2$, т. е.

$$\lim_{n \rightarrow \infty} \frac{1}{n} (x_1 + \dots + x_n) = \frac{1}{2},$$

б) в подпоследовательности $x_i = \{x_{i_1}, x_{i_2}, \dots\}$ 0—1 последовательности $x = \{x_1, x_2, \dots\}$, определяемой любым «правилом выбора», относительная частота появления 1 равна $1/2$, т. е.

$$\lim_{n \rightarrow \infty} \frac{1}{n} (x_{i_1} + x_{i_2} + \dots + x_{i_n}) = \frac{1}{2}.$$

При этом вероятности определяются как предел частот. У Мизеса, однако, возникли трудности при строгом определении понятия «правило выбора». Его идеи не выдержали проверки временем. Были построены примеры (Вилль, 1939) 0—1 последовательностей, удовлетворяющих требованиям Р. Мизеса к случайным последовательностям, но не удовлетворяющих некоторым теоремам теории вероятностей (в частности, теореме о повторном логарифме

$$\limsup_{n \rightarrow \infty} \frac{x_1 + \dots + x_n - \frac{n}{2}}{\frac{1}{2} \sqrt{2n \log \log n}} = 1 \text{ и}$$

$$\liminf_{n \rightarrow \infty} \frac{x_1 + \dots + x_n - \frac{n}{2}}{\frac{1}{2} \sqrt{2n \log \log n}} = -1 \Bigg).$$

Дело здесь, грубо говоря, в том, что Р. Мизес, определяя случайность как отсутствие закономерностей, учитывал только частотные закономерности. С точки зрения современного подхода А. Н. Колмогорова и П. Мартин-Лефа к случайности, этого оказывается недостаточно.

В 1933 г. А. Н. Колмогоров предложил рассматривать теорию вероятностей как прикладную теорию меры. Элегантность и формальная завершенность построений Колмогорова привлекли специалистов по теории вероятностей, и развитие идей Р. Мизеса затормозилось. Однако плата за новый подход к теории вероятностей оказалась существенной. Пришлось отказаться от рассмотрения случайных индивидуальных (бесконечных) последовательностей и рассматривать только случайные множества последовательностей.

2. В 1965 г. П. Мартин-Леф, основываясь на идеях А. Н. Колмогорова, дал определение случайной последовательности, свободное от трудностей, встретившихся в концепции Р. Мизеса. «Не случайными» предлагается считать только те последовательности, в которых наблюдается достаточно много закономерностей. При этом под закономерностью подразумевается любое проверяемое свойство последовательности, присущее лишь узкому классу последовательностей.

Пусть $x = \{x_1, \dots, x_n, \dots\}$ бесконечная двоичная последовательность. Рассматривая последовательно удлиняющиеся начальные фрагменты этой последовательности $(x)_n = \{x_1, \dots, x_n\}$, будем пытаться устанавливать скрытые в них закономерности — правила, позволяющие их, по крайней мере грубо, восстановить.

Традиционный подход к выявлению закономерностей бесконечной последовательности связан с простыми процедурами подсчета частот, с которыми встречаются 0 и 1 во фрагментах двоичной последовательности. Однако интуитивно понятие «случайность» противостоит не столько жесткой повторяемости, сколько более широкому понятию «организованности». В свою очередь, степень организованности естественно характеризовать сложностью алгоритма, позволяющего восстанавливать последовательно удлиняющиеся начальные фрагменты бесконечной последовательности. Ясно, что определение степени организованности бесконечной последовательности не сводится только к подсчету частот.

П. Мартин-Леф формализовал идею выделения закономерностей и предложил подход к оценке «количества закономерностей», содержащихся в бесконечной последовательности.

Под «тестом на случайность» будем понимать вычислимую функцию F на множестве конечных двоичных после-

довательностей со значениями из натурального ряда, удовлетворяющую следующему условию: для каждого $m > 0$ для множества последовательностей x , у которых существует начальный фрагмент $(x)_n$ с $F((x)_n) \geq m$, не превосходит 2^{-m} , т. е. доля множества тех последовательностей, в которых может быть обнаружено более чем на m бит закономерностей, не должна превосходить 2^{-m} .

Обозначим через $F(x)$ верхнюю грань по n значений $F((x)_n)$ и будем говорить, что последовательность x «выдерживает тест F », если $F(x)$ конечно, или, что «тест F отбрасывает x », если $F(x) = \infty$. Значение $F(x)$ теста есть «количество» найденных им закономерностей. Функция F показывает, на сколько бит закономерностей типа, определяемого функцией F , можно установить в последовательности x . Каждый бит закономерностей должен по крайней мере вдвое сократить множество допустимых последовательностей.

Двоичная последовательность называется **случайной**, если она выдерживает все тесты. При таком определении случайной последовательности можно доказать, что они удовлетворяют всем эффективно проверяемым законам теории вероятностей (закону больших чисел, закону повторного логарифма и т. д.). В случае если какой-либо вероятностный закон не выполняется, его условия индуцируют некоторую вычислимую функцию — тест, который не выдерживается проверяемыми последовательностями (невыполнение какого-либо вероятностного закона есть закономерность).

Требование вычислимости F обеспечивает передачу проверки на случайность вычислительной машине.

3. По определению последовательность может быть, строго говоря, объявлена случайной после прохождения ею всех тестов. Стало быть, установление числа бит закономерностей в бесконечной двоичной последовательности производится тем точнее, чем больше вычислимых (частично-рекурсивных) функций использовано в качестве тестов. Казалось бы, проверка последовательности на случайность и оценка числа бит закономерностей в ней совсем не конструктивная процедура. Оказывается, это не совсем так. Как мы видели, переход к оптимальным (универсальным) структурам позволяет ослабить зависимость алгоритмической сложности или сложности объекта от «структуры» вычислительной машины (от выбранной частично-рекурсивной функции). Аналогичным образом можно заменить

проверку на случайность по всем тестам (по всем частично-рекурсивным функциям) проверкой по одной так называемой универсальной функции (по одному универсальному тесту). П. Мартин-Леф доказал, что существует тест F^* (называемый универсальным) такой, что для любого другого теста F при всех бесконечных последовательностях x выполняется неравенство $F(x) \leq F^*(x) + C$, где C — некоторая константа, не зависящая от проверяемой последовательности x . Теорема П. Мартин-Лефа позволяет уточнить определение случайной последовательности.

Бесконечная двоичная последовательность случайна, если она выдерживает универсальный тест.

4. Конечно, замена проверки на случайность по всем тестам проверкой по одному — универсальному — не дается даром и установление случайного характера бесконечной последовательности не становится от этого существенно более конструктивной операцией. Доказательство существования универсального теста имеет главным образом теоретическое значение. Оно уточняет понятие «случайность» и позволяет получить ряд качественных выводов.

Обозначим через $p(x)$ число бит закономерностей двоичного слова x , которые найдены в нем универсальным тестом F^* ; $l(x)$ — по-прежнему длина слова x . По определению для вычисления $p(x)$ следует:

а) применить F^* ко всем последовательностям $(\xi)_n$ длины n , начинающимся с фрагмента x ;

б) вычислить наименьшее значение F^* на таких последовательностях $(\min_{(\xi)_n} F^*((\xi)_n))$;

в) взять максимум полученной величины по n

$$p(x) = \max_n \min_{(\xi)_n} F^* \{ ((\xi)_n) \mid \text{fragment}_0(\xi)_n = x \}.$$

Фундаментальный результат Мартин-Лефа, устанавливающий связь между понятиями случайности и сложности, формулируется в следующем утверждении.

Сумма сложности слова x и числа бит его закономерностей почти совпадает с длиной двоичной записи этого числа. Точнее,

$$|(p(x) + K(x)) - l(x)| \leq 4l(l(x)) + \text{const}.$$

Если последовательность $\xi = \{\xi_1, \dots, \xi_n, \dots\}$ случайна, то отсюда следует, что сложность начального фрагмента $(\xi)_n$ длины n

$$K((\xi)_n) \geq n - 4l(n) + \text{const} \quad \forall n.$$

Полученное неравенство означает, что сложность (достаточно длинных) начальных фрагментов бесконечной случайной последовательности приблизительно равна их длине. Это важный результат, поскольку он позволяет свести анализ бесконечной случайной последовательности к изучению конечных последовательностей, каждая из которых может быть построена с помощью подходящего алгоритма.

Последнее неравенство может быть также использовано в качестве хоть и грубого, но содержательного определения индивидуальной случайной последовательности. Случайной следует признать всякую последовательность, у которой алгоритмическая сложность начального фрагмента (при подходящем (оптимальном) способе его описания) растет достаточно быстро с увеличением длины фрагмента.

Алгоритмическая сложность и теория информации

1. Отправляясь от алгоритмической сложности, можно показать, что установившийся подход к теории информации, основанный на принципах теории вероятностей, не является единственным и наиболее естественным. Собственно, у колыбели теории алгоритмической сложности находится работа А. И. Колмогорова о новом (алгоритмическом) подходе к понятию «количество информации» [22].

Важнейшим понятием теории информации является «энтропия» сообщений. Привычная шенноновская трактовка энтропии — это мера неопределенности результатов опыта, могущего иметь в зависимости от случая различные исходы. Приведем умозрительные основания для выбора энтропии в качестве меры неопределенности.

Пусть опыт A имеет s равновероятных исходов, а опыт B — l равновероятных исходов. В этом случае степень F неопределенности результатов опыта зависит только от числа исходов: $F_A = F(s)$, $F_B = F(l)$. Число исходов опыта AB (т. е. объединенных в один опытов A и B) равно sl . Естественно полагать, что мера неопределенности опыта AB равна сумме мер неопределенности опытов A и B , т. е. $F(sl) = F(s) + F(l)$. По определению опыт, имеющий единственный исход, не связан с неопределенностью. Его степень неопределенности $F(1) = 0$. Ясно также, что степень неопределенности должна расти с увеличением числа исходов, т. е. $F(s) > F(l)$ при $s > l$. Перечисленным свойствам

удовлетворяет функция $F(s) = \log s$. Можно показать, что с точностью до нормирующего множителя это единственная функция, удовлетворяющая сформулированным требованиям к мере неопределенности равновероятных событий ².

Таким образом, если исходы опыта равновероятны и реализуются каждый с вероятностью $p = \frac{1}{s}$, то общая мера неопределенности опыта равна $\log s = -\log \frac{1}{s} = -\log p$, а мера неопределенности каждого исхода равна $\frac{1}{s} \log s = -p \log p$.

Пусть теперь исходы опыта неравновероятны и производится m независимых испытаний с s возможными исходами с вероятностями p_1, p_2, \dots, p_s соответственно. Продолжая приведенные выше интуитивные рассуждения, естественно принять в качестве меры неопределенности i -го исхода — $p_i \log p_i$, а в качестве общей меры неопределенности опыта — сумму $\sum_{i=1}^s p_i \log p_i$. Эта величина называется энтропией опыта A и обозначается $H(A)$ или $H(p)$. К. Шеннон [43] приводит набор естественных требований к мере неопределенности опыта с неравновероятными исходами. Энтропия является единственной с точностью до нормирующего множителя функцией, удовлетворяющей этим требованиям.

2. А. Н. Колмогоров [23] приводит другое (отличное от меры неопределенности результатов опыта) толкование энтропии. Его можно аргументировать как вероятностными, так и комбинаторными соображениями. При этом комбинаторная аргументация связана с существенно более слабыми допущениями.

Пусть по-прежнему проводится m независимых испытаний с s возможными исходами, наступающими соответственно с вероятностями p_1, \dots, p_s . Пусть m достаточно велико. Оказывается, что для записи исходов всей серии испытаний необходимо приблизительно mH битов. Вероятностная аргументация этого утверждения существенно использует схему независимых испытаний. Тем не менее это утверждение может быть доказано при гораздо менее жестких предположениях чисто комбинаторного порядка, не

²) Мы предполагаем основание логарифмов равным 2. Тогда степень неопределенности будет измеряться в битах (в двоичных единицах).

связанных с вероятностными представлениями о независимости испытаний.

Пусть i -й исход, $i=1, \dots, s$, в результате испытаний реализован m_i раз. Перенумеруем все возможные расположения исходов (их число $C(m_1, \dots, m_s) = \frac{m!}{m_1! \dots m_s!}$). Можно зафиксировать результат испытаний, указав количества m_1, \dots, m_s каждого из исходов и номер той последовательности исходов, которая имела место в процессе испытаний. Чтобы записать количество реализаций каждого исхода, требуется не более $s \log m$ бит информации (поскольку всего исходов s , а запись m_i , $i=1, \dots, s$, содержит не больше двоичных знаков, чем запись числа испытаний m , т. е. не превышает $\log m$). Номер реализованной последовательности исходов не превышает $C(m_1, \dots, m_s)$. Запись C требует $\log C$ двоичных знаков. Таким образом, для записи последовательности исходов при m испытаниях необходимо не более $s \log m + \log C$ бит информации.

Используя формулу Стирлинга $m! = \sqrt{2\pi m} \left(\frac{m}{e}\right)^m e^{\frac{\mu_m}{12m}}$, где $|\mu_m| \leq 1$, получаем, что при большом m запись последовательности исходов испытаний требует порядка

$$m \left(- \sum_{i=1}^s \frac{m_i}{m} \log \frac{m_i}{m} \right) \sim mH \text{ двоичных знаков.}$$

Заметим, что в приведенных рассуждениях не было речи о независимости испытаний и о вероятностях реализаций исходов. Полученный вывод логически не зависит от каких бы то ни было вероятностных допущений.

При независимых испытаниях и большом m в силу закона больших чисел $\frac{m_i}{m} \sim p_i$ и оценка длины записи последовательности исходов испытаний в комбинаторном подходе та же, что и в вероятностном, использующем существенно более жесткие допущения.

Аналогичным образом можно, исходя из комбинаторных соображений, получить энтропию цепи Маркова при существенно более слабых предположениях, чем при вероятностном подходе.

3. В классической (шенноновской) теории информации количество информации, содержащееся в объекте x об объекте y , определяется через энтропию $H(y)$ объекта y и условную энтропию $H(y|x)$:

$$I(y|x) = H(y) - H(y|x).$$

При этом энтропия (условная и безусловная) определяется из вероятностных соображений. Комбинаторная трактовка энтропии $H(y)$ (аналогичные рассуждения могут быть проведены и относительно условной энтропии $H(y|x)$) и приведенные выше соображения о мажорантах сложности позволяют дать не связанное с вероятностными предположениями толкование понятия «количество информации».

Сложность $K(y)$ объекта y можно интерпретировать как информацию, содержащуюся в объекте y о самом себе, т. е. как количество информации, необходимое для восстановления объекта y . Аналогичным образом в качестве количества информации, которую следует добавить к информации, содержащейся в x , чтобы восстановить y , естественно рассматривать условную (алгоритмическую) сложность $K(y|x)$. Разность между этими величинами целесообразно принять за определение количества информации в x об y :

$$I_{\Phi}(y|x) = K_{\Phi}(y) - K_{\Phi}(y|x).$$

Индекс Φ здесь указывает «структуру» вычислительной машины, относительно которой оценивается сложность.

Классическое определение количества информации естественно для массовых явлений, описываемых вероятностными характеристиками (например, для оценки информации, содержащейся в потоке телеграмм). Шенноновский подход приспособлен к требованиям теории передачи по каналам связи массовой информации, состоящей из большого числа слабо связанных сообщений, удовлетворяющих некоторым статистическим закономерностям. Вероятностное определение количества информации не поддается истолкованию применительно к индивидуальным событиям. Неясно, например, как в понятиях традиционной теории информации оценить количество информации, содержащееся в переводе некоторой книги на другой язык, относительно оригинала.

Колмогоровское определение количества информации относится к индивидуальным объектам и в принципе дает возможность оценить содержащуюся в них информацию. Однако практические аспекты приложения количественных методов оценки информации, содержащейся в индивидуальных объектах, ждут еще своего исследования и разрешения. Они должны учитывать формальные особенности, связанные с определением алгоритмической сложности. В частности, в каждом отдельном случае следует аргумен-

тировать выбор универсальной структуры, относительно которой оценивается сложность. Ведь в отличие от шенноновского определения количества информации колмогоровское определение сохраняет некоторую неопределенность и зависит от выбора универсальной структуры φ . Разные варианты $I_{\varphi}(y|x)$ и $I_{\psi}(y|x)$ комбинаторного определения количества информации эквивалентны лишь в смысле

$$|I_{\varphi}(y|x) - I_{\psi}(y|x)| \leq C_{\varphi\psi},$$

где константа $C_{\varphi\psi}$ зависит от универсальных структур, относительно которых определяется алгоритмическая сложность.

4. Комбинаторные определения энтропии и количества информации заставили пересмотреть ряд утверждений теории информации.

В частности, вместо привычных утверждений шенноновской теории информации, основанной на вероятностном подходе,

$$\begin{aligned} H(x, y) &= H(x) + H(y|x), \\ I(y|x) &= I(x|y) \end{aligned}$$

при комбинаторном подходе имеют место уточненные соотношения

$$\begin{aligned} H(x, y) &= H(x) + H(y|x) + O(\log K(x, y)), \\ |I(y|x) - I(x|y)| &= O(\log K(x, y)). \end{aligned}$$

При этом установлено, что дефект в коммутативности при комбинаторном определении количества информации действительно может иметь указанный в последней формуле порядок.

Содержательный и формальный анализ показывает, что применение уточненных соотношений для энтропии и количества информации к индивидуальным объектам приводит к более естественным выводам, чем использование утверждений классической теории информации, а в приложениях, доступных вероятностному подходу, расхождения между выводами, вытекающими из колмогоровского и шенноновского определений количества информации, пренебрежимо малы.

Сложность табулирования

1. К проблеме алгоритмической сложности примыкает задача оценки сложности табулирования функций. Работам А. Н. Колмогорова по алгоритмической сложности предшествовали его работы и работы его учеников А. Г. Витушкина и В. И. Арнольда по XIII проблеме Гильберта, которая связана с оценкой сложности функций, в частности, в интересах их экономного табулирования. Можно думать, что эти работы стимулировали проблематику алгоритмической сложности.

В 1900 г. на Втором международном математическом конгрессе Д. Гильберт выдвинул 23 нерешенные проблемы, которые, с его точки зрения, определяли направления развития математики XX века. XIII проблема формулировалась следующим образом: существуют ли аналитические функции трех переменных, не представимые суперпозицией непрерывных функций меньшего числа переменных. Исходя из этого, он предположил, что «сложность» функций определяется главным образом числом аргументов. Однако, как выяснилось позднее, эта гипотеза, справедливая для аналитических функций, неверна в общем случае для более широких классов функций. В частности, для s раз дифференцируемых функций n переменных характеристикой сложности функций оказывается не число ее аргументов, а отношение $\frac{n}{s}$. (Чтобы больше не возвращаться к XIII проблеме Гильберта, заметим, что окончательное ее решение и опровержение гипотезы Д. Гильберта принадлежит А. Н. Колмогорову и В. И. Арнольду. В 1956 г. Колмогоров доказал, что всякая непрерывная функция n переменных может быть представлена в виде суперпозиции непрерывных функций трех переменных. В 1957 г. В. И. Арнольд — тогда еще студент мехмата МГУ — доказал, что всякая непрерывная функция трех переменных представима в виде суперпозиции непрерывных функций двух переменных. В том же 1957 г. А. Н. Колмогоров поставил точку над XIII проблемой Гильберта. Он доказал, что любая непрерывная функция n переменных может быть представлена в виде суперпозиции непрерывных функций одного переменного и операции сложения).

Исследования А. Н. Колмогорова и А. Г. Витушкина, стимулированные XIII проблемой Гильберта, естественным образом привели к оценкам объема наиболее экономных

таблиц для записи функций разных классов (функций, принадлежащих тем или иным функциональным пространствам) — к оценкам общего количества двоичных знаков, необходимых для записи всех параметров таблицы, и к оценке сложности алгоритма, расшифровывающего таблицу. Заметим, что, несмотря на прогресс, достигнутый в последние годы в разработке запоминающих устройств большой емкости, необходимость в составлении экономных таблиц отнюдь не отпала. Огромные объемы информации, подлежащие запоминанию в современных экономических, социальных, технических и военных системах, и дефицит времени на поиск нужных данных сохраняют актуальность проблемы экономного табулирования.

2. Приведем нестрогую (упрощенную) постановку задачи об оценке сложности таблиц функций заданного класса.

Пусть F — некоторое семейство функций $f(x)$, определенных на множестве $G \subset E^n$, а $T_\varepsilon(f)$ — таблица любой функции $f(x) \in F$, состоящая из значений p параметров y_1, y_2, \dots, y_p и расшифровывающего алгоритма $A(y)$, восстанавливающего $f(x)$ с точностью до ε . Часто под алгоритмом $A(y)$ понимают полином $P_x^k(y)$ от p переменных y_1, y_2, \dots, y_p степени не выше k по каждому из переменных, коэффициенты которого произвольным образом зависят от $x \in G$, такой, что при всех $f(x) \in F$ существует набор $y = (y_1, \dots, y_p)$, на котором

$$|f(x) - P_x^k(y)| \leq \varepsilon \quad \forall x \in G.$$

Задача заключается в том, чтобы по свойствам класса F функций $f(x)$ оценить «сложность» таблиц для элементов из F . Сложность таблицы определяется, во-первых, ее объемом — минимальным числом двоичных знаков, необходимых для записи всех параметров таблицы, и, во-вторых, сложностью расшифровывающего алгоритма. В частности, если под $A(y)$ понимают полином $P_x^k(y)$, то сложность расшифровывающего алгоритма определяется числами p и k .

Легко сообразить, что наиболее экономная запись в таблицу элемента x из конечного множества $X = \{x_1, \dots, x_n\}$ содержит $\log_2 n$ двоичных знаков. Величина $\log_2 n$ называется энтропией конечного множества $X = \{x_1, \dots, x_n\}$ и обозначается $H(X)$. Если X состоит из бесконечного числа элементов, то записать любой его элемент с помощью конечного числа двоичных знаков невозможно. Для оценки объе-

ма таблиц, отвечающих бесконечным множествам, необходимо объединить близкие по своим свойствам элементы в группы так, чтобы число групп было конечным.

В 1955 г. А. Н. Колмогоров ввел понятие ϵ -энтропии метрического пространства F функций. Под ϵ -энтропией пространства F функций понимают величину

$$H_{\epsilon}(F) = \log_2 N_{\epsilon}(F),$$

где $N_{\epsilon}(F)$ — число элементов минимальной в F по числу точек ϵ -сети множества F . (Напомним, что множество $Y \subset F$ называется ϵ -сетью пространства F , если для всех $f \in F$ можно указать $y \in Y$ такой, что $\rho_F(y, f) \leq \epsilon$. Здесь $\rho_F(y, f)$ — расстояние между y и f в метрике F .)

ϵ -энтропия какого-либо класса функций — это, грубо говоря, количество информации, позволяющее выделить функцию этого класса с точностью ϵ . А. Н. Колмогоров показал, что ϵ -энтропия s раз дифференцируемых функций n

переменных растет как $\epsilon^{-\frac{n}{s}}$. Это значит, что количество информации, необходимое для того, чтобы восстановить функцию класса, тем больше, чем больше n — число независимых переменных, и тем меньше, чем выше гладкость s .

При этом существенно именно отношение $\frac{n}{s}$. Отсюда, между прочим, следует теорема Витушкина: при всяких n и s существует s раз дифференцируемая функция n переменных, не представляемая суперпозициями p раз дифференцируемых функций m переменных, если только $\frac{m}{p} < \frac{n}{s}$. Ведь количество информации, необходимое для задания суперпозиции, не может превосходить (по порядку величины) количество информации, достаточное для задания составляющих суперпозицию меньшего числа переменных.

А. Г. Витушкин расширил колмогоровское понятие ϵ -энтропии и ввел понятие относительной ϵ -энтропии. Он разрешил записывать в таблицы функций из класса F функции, хорошо аппроксимирующие элементы из F , но не принадлежащие F . Таким образом, он получил возможность строить более экономные методы составления таблиц.

Пусть Φ — метрическое расширение пространства F (т. е. Φ содержит F и при всех $f_1, f_2 \in F$ $\rho_F(f_1, f_2) = \rho_{\Phi}(f_1, f_2)$, где ρ_F и ρ_{Φ} — расстояния между f_1 и f_2 в смысле метрик F и Φ соответственно).

Под ϵ -энтропией пространства F функций относительно

его метрического расширения Φ (или просто под относительной ε -энтропией F) понимают величину

$$H_{\varepsilon}^{\Phi}(F) = \log_2 N_{\varepsilon}^{\Phi}(F),$$

где $N_{\varepsilon}^{\Phi}(F)$ — число элементов минимальной в Φ по числу точек ε -сети множества F .

Ясно, что колмогоровская ε -энтропия пространства F функций $H_{\varepsilon}(F) = H_{\varepsilon}^F(F)$.

Будем называть **таблицей** $T_{\varepsilon}^{\Phi}(f)$ функций $f(x)$ из F , восстанавливающей f с точностью до ε при помощи некоторой функции φ из Φ , набор $y = (y_1, \dots, y_p)$ элементов некоторого множества Y и алгоритм $A(y)$, который ставит в соответствие набору y некоторый элемент $\varphi \in \Phi$ такой, что $\rho_{\Phi}(\varphi, f) \leq \varepsilon$. Алгоритм $A(y)$ строит в пространстве Φ ε -сеть для F .

Объемом таблицы $T_{\varepsilon}^{\Phi}(f)$ называется число $P(T_{\varepsilon}^{\Phi}(f)) = p \log_2 n$, где n — число элементов множества Y .

Имеет место следующая оценка относительной ε -энтропии класса F функций:

$$\inf_{T_{\varepsilon}^{\Phi}(f)} P[T_{\varepsilon}^{\Phi}(f)] \leq H_{\varepsilon}^{\Phi}(F) \leq \inf_{T_{\varepsilon}^{\Phi}(f)} P[T_{\varepsilon}^{\Phi}(f)] + 1.$$

Таким образом, при фиксированном Φ для $f \in F$ относительная ε -энтропия $H_{\varepsilon}^{\Phi}(F)$ совпадает с точностью до 1 с нижней гранью объема таблицы $T_{\varepsilon}^{\Phi}(f)$.

Наметим схему алгоритма сопоставления с точностью до ε функции $f \in F$ таблицы минимального объема. Фиксируем в расширенном пространстве Φ минимальную ε -сеть пространства F . Обозначим ее элементы через $\varphi_1, \varphi_2, \dots, \varphi_n$ ($n = N_{\varepsilon}^{\Phi}(F)$). Примем в качестве параметра таблицы для $f \in F$ номер того элемента φ , который аппроксимирует f с точностью ε . Поскольку номер φ меняется от 1 до n , то его двоичная запись занимает $[\log N_{\varepsilon}^{\Phi}(F)] + 1 = [H_{\varepsilon}^{\Phi}(F)] + 1$ двоичных знаков.

3. Назовем абсолютной ε -энтропией $\tilde{H}_{\varepsilon}(F)$ метрического пространства F нижнюю грань относительных ε -энтропий F при всевозможных его метрических расширениях Φ

$$\tilde{H}_{\varepsilon}(F) = \inf_{\Phi \supset F} H_{\varepsilon}^{\Phi}(F).$$

Оказывается, нижняя грань достигается на пространстве S всех непрерывных функций. Таким образом, пространство S является наилучшим метрическим расширением для всякого (компактного) метрического пространства F (с равномерной метрикой)

$$H_\varepsilon^C(F) = \bar{H}_\varepsilon(F) \leq H_\varepsilon^\Phi(F) \quad \forall \Phi.$$

Прикладной смысл абсолютной ε -энтропии определяется следующим утверждением.

Объем таблицы $T_\varepsilon^\Phi(f)$ функции $f \in F$ не меньше абсолютной ε -энтропии $\bar{H}_\varepsilon(F)$. При этом для каждой функции $f \in F$ можно составить таблицу, объем которой не больше $[\bar{H}_\varepsilon(F)] + 1$.

Доказательства приведенных утверждений содержатся в монографии [9]. Там же приведены, кроме того, неравенства — оценки сложности таблиц для ряда функциональных пространств. Приведем здесь лишь некоторые оценки параметров p и k наиболее экономных алгоритмов, восстанавливающих функции $f \in F$ с точностью до ε с помощью таблиц $T_\varepsilon^\Phi(F)$.

$$1) \quad p \log \frac{k+1}{\varepsilon} \geq \alpha_F \bar{H}_\varepsilon(F),$$

где α_F — некоторая константа, определяемая классом функций F и не зависящая от p , k и ε .

2) Существуют методы составления таблиц $T_\varepsilon^\Phi(f)$, для которых

$$p \log \frac{k+1}{\varepsilon} \leq \beta_F \bar{H}_\varepsilon(F),$$

где β_F — некоторая константа, зависящая лишь от класса F табулируемых функций. В частности, к таким методам составления таблиц относятся методы, основанные на запоминании коэффициентов фрагмента ряда Тейлора функций.

Для некоторых классов F функций в [9] получены приближенные выражения для абсолютной ε -энтропии F . Они позволяют уточнить приведенные оценки параметров сложности таблиц функций соответствующих классов.

К сожалению, классы F , для которых получены оценки сложности таблиц, выделены сугубо формальными свойствами функций. На практике обычно возникает потребность

в составлении таблиц функций, отвечающих реальным явлениям и процессам и классифицируемых по содержательным признакам. Можно, однако, полагать, что рабочий аппарат, изложенный в [9], окажется полезным для оценки сложности соответствующих таблиц и для разработки методов наиболее экономного использования памяти вычислительных машин.

Сложность и закон необходимого разнообразия

1. К законам кибернетики часто относят закон необходимого разнообразия У. Р. Эшби [48]. Как и многие другие утверждения, декларировавшиеся на ранней стадии развития кибернетики, этот закон сформулирован в недостаточно четких терминах и апеллирует к интуиции читателя. Одна из формулировок закона необходимого разнообразия (которая, скорее, навеяна духом работ У. Р. Эшби, чем вытекает из его рассуждений) такова: для нормального функционирования управляемой системы, обеспечивающего полное использование ее потенциала, требуется, чтобы разнообразие управляющей системы было не меньше разнообразия управляемого объекта. При этом понятия «нормальное функционирование» и «разнообразие» не уточняются и в различных работах толкуются по-разному. Различные авторы интерпретируют термин «нормальное функционирование» как «устойчивое поведение» или «пребывание в состоянии гомеостазиса» или «целеустремленное развитие». Термин «разнообразие» толкуется как «число степеней свободы», или как «сложность», так или иначе определенная на эвристическом уровне. Сам У. Р. Эшби — по специальности биолог, чуждый формально-логического способа выражения своих мыслей, — и не пытался уточнить основные понятия закона необходимого разнообразия, интуитивно (и, наверное, справедливо) полагая, что в каждой конкретной области профессионал это сделает лучше, а приемлемая для всех (или для многих) достаточно строгая и в то же время содержательная формулировка закона, если она вообще возможна, отшлифуется со временем.

Проявления закона Эшби встречаются на каждом шагу. Качественные выводы из закона необходимого разнообразия естественны и не слишком чувствительны к уточнению понятий, используемых в формулировке закона. Ясно, например, что если объект управления обладает большим

разнообразием — многими состояниями равновесия, позволяющими приспосабливаться к возможным возмущениям среды, то рациональная управляющая система не должна ограничивать потенциальных возможностей объекта. Структура и производительность (разнообразие) управляющей системы должны обеспечить отслеживание поведения среды и адаптацию объекта к состоянию среды.

Закон Эшби стихийно проявляется и в проблемах организации. Чтобы исключить конфликтные ситуации в организационных системах, необходимо, в частности, определенное соответствие между возможностями (разнообразием) коллектива и руководителя. Управляющий должен обладать качествами, позволяющими использовать потенциал коллектива. В противном случае следует либо увеличить разнообразие управления — заменить руководителя более подходящим, ввести промежуточное звено, либо, если это по каким-то причинам невозможно или нецелесообразно, сократить разнообразие объекта — «заглубить» коллектив — перевести членов коллектива, чей потенциал полностью не используется, в другое (желательно более подходящее) подразделение. Как видно, в данном случае закон необходимого разнообразия вполне согласуется с первым законом Паркинсона.

Приведем еще пример стихийного проявления закона необходимого разнообразия.

Вся сознательная жизнь человека представляет собой непрерывную цепь выбора решений. Разрешить возникающую задачу — значит, изменить ситуацию желаемым образом. Чтобы человек был в состоянии это сделать, он должен обладать большим разнообразием, чем ситуация, которую он старается изменить. Разнообразие лица, принимающего решение, определяется его физическими и материальными возможностями и моральными принципами — ограничениями на выбор средств достижения цели. Разнообразие ситуации определяется факторами, подлежащими учету, и трудоемкостью их изменения в желаемом направлении. Задача оказывается неразрешимой для лица, принимающего решение, если его разнообразие не соответствует разнообразию ситуации. В поисках путей преодоления противоречий человек пытается сократить разнообразие ситуации — упростить постановку задачи, умерить свои желания или повысить свое разнообразие (расширить свои материальные возможности или снизить ограничения, определяемые моральными принципами). К сожалению, не-

редкий случай, когда разрешимость задачи достигается «наиболее экономным путем» — за счет пренебрежения моральными ограничениями. Из медицинской практики известно, однако, что преодоление противоречий за счет уступок в принципах может приводить к психическим срывам.

2. На практике далеко не всегда можно рассчитывать на стихийное проявление закона необходимого разнообразия при установлении рационального соответствия между объектом управления и управляющей системой. Обоснованное проектирование управляющих устройств для сложных объектов требует четкого определения условий согласования и, в частности, понятия «разнообразие».

Известны работы, в которых предлагается основывать согласование управления и объекта по соответствию между алгоритмическими сложностями объекта управления и системы переработки информации, реализуемой управляющим устройством.

Объект и система управления характеризуются различными параметрами. Параметры объекта определяются его возможными состояниями и механизмом переходов из состояния в состояние и существенно зависят от целевого назначения объекта. Параметры управляющего устройства определяются его структурой, объемами различных типов памяти, скоростью выполнения тех или иных операций. Объект управления создает рабочую нагрузку для управляющего устройства.

В качестве эффективной меры рабочей нагрузки предлагается использовать сложность объекта. Управляющий вычислительный комплекс не должен снижать потенциал управляемого объекта. Его сложность поэтому не должна быть меньше сложности объекта. Таким образом, алгоритмическая сложность выступает здесь как мера разнообразия. Выбор системы управления сводится, следовательно, к выбору наиболее экономного вычислительного комплекса, сложность которого не меньше сложности управляемого объекта.

Для практического использования подобного подхода необходимо связать сложность объекта и сложность вычислительного комплекса (точнее, их оценки) с технико-экономическими параметрами управляемого объекта и системы управления и учесть, таким образом, соответствие между логическими моделями объекта и управления и их физическими образами. Это трудоемкая задача, которая вряд ли может быть решена без продуманной идеализации объекта

и системы управления (например, в виде конечных автоматов) и рационально организованного имитационного моделирования. Трудоемкость решения задачи обусловлена, кроме того, тем, что алгоритмическая сложность — невычислимая функция. Поэтому возникают дополнительные проблемы, связанные с выбором подходящей мажоранты сложности или с сужением класса рассматриваемых систем. В частности, исследования упрощаются, если ограничиться системами с заданными параметрами.

Известны некоторые небезуспешные попытки установления таким образом соответствия вычислительной системы и объекта управления. Они позволили выявить перегруженные звенья управляющей системы и прогнозировать пути развития системы по мере эволюции объекта.

Законы сохранения (массы, энергии, импульса) или связанные с ними вариационные принципы в задачах естественных наук определяют область допустимых изменений процессов неживой природы. Из множества возможных движений реализуются лишь те, которые оптимизируют некоторые функционалы или попадают в некоторые области, допускаемые соответствующими законами сохранения. Законы сохранения, таким образом, участвуют в формировании моделей явлений и процессов в науках о неживой природе. В моделях принятия решений — в планировании, управлении, проектировании и организации — область допустимых решений определяется обычно балансовыми соотношениями. Дополнительные неравенства, устанавливаемые законом необходимого разнообразия, сужают область рациональных решений, выделяя из нее только те решения, которые гарантируют согласованность объекта управления и системы управления.

3. В некоторых работах по сложным динамическим системам под нормальным функционированием подразумевают поведение, обеспечивающее пребывание управляемого объекта в гомеостатическом состоянии, т. е. в состоянии, при котором существенные характеристики объекта не выходят из некоторой благоприятной для его жизнедеятельности области. Такие системы называют гомеостатическими.

Гомеостатические системы состоят из пассивной и активной подсистем — из собственно объекта и управляющего устройства. Управляющее устройство возвращает систему в гомеостатическое состояние при небольших возмущениях объекта, вызываемых средой обитания системы. Таким образом, активная подсистема обеспечивает сохранение гомео-

стазиса системы в заданном диапазоне изменения характеристик среды. В соответствии с законом Эшби разнообразие активной подсистемы (АП) должно быть не меньше разнообразия пассивной подсистемы (ПП). Под разнообразием управления и объекта подразумевается в данном случае подходящая для изучаемой конкретной ситуации компонента сложности активной подсистемы ($C_{\text{ап}}$) и пассивной подсистемы ($C_{\text{пп}}$). Чтобы система при ожидаемых изменениях среды не выходила из состояния гомеостаза, необходимо, чтобы было $C_{\text{ап}} \geq C_{\text{пп}}$. Выражая $C_{\text{ап}}$ и $C_{\text{пп}}$ через параметры, подлежащие регулированию, получаем ограничения на их выбор, вытекающие из закона необходимого разнообразия.

Проблема поддержания гомеостаза приобретает сейчас особый интерес в связи с экологическими задачами.

Природа и люди представляют собой единую систему, в которой на разных этапах эволюции неживая природа и человеческое общество являются пассивной или активной подсистемой. Состояние гомеостаза, в котором находится система, включающая человека и биосферу, определяет благоприятные условия для жизни. Человек в процессе производственной деятельности меняет параметры биосферы и сам непрерывно адаптируется к изменяющимся условиям существования, сохраняя при этом гомеостаз системы. Но приспособительные возможности человека и природы ограничены. Рассогласование в развитии биосферы и человеческого общества чревато нарушением гомеостаза системы и может подорвать основу современной цивилизации.

Производственная деятельность человека оказывает на природу как разрушительное, так и восстанавливающее действие. Поддержание гомеостаза требует превышения разнообразия восстанавливающих процессов над разнообразием разрушительных действий. Разнообразие восстановительных механизмов должно превышать разнообразие процессов, разрушающих природу. Разработка моделей взаимодействия природы и общества и методов решения возникающих здесь проблем — основная задача экологии — науки о собственном доме.

Постановка экологических проблем должна быть согласована с динамикой эволюционных процессов. Конечно, основная угроза равновесию между природой и обществом связана с опасностью ядерного конфликта. Однако следует иметь в виду, что не только ядерная катастрофа,

но и незаметные изменения антропогенной нагрузки на биосферу могут вывести траекторию эволюционного процесса в область, не приемлемую для жизнедеятельности человека. Неконтролируемое развитие производства грозит существованию человеческого рода.

Одной из актуальнейших задач человечества становится управление эволюционными процессами в масштабе планеты. Координация развития общества и окружающей среды требует согласованных усилий специалистов естественных и общественных наук.

Следует полагать, что в формировании теории управления эволюционными процессами и в разработке количественных и качественных методов анализа глобальных решений далеко не последнюю роль будут играть интенсивно развивающиеся сейчас методы оценки различных компонент сложности явлений и процессов, определяющих разрушительные и восстановительные воздействия человеческого общества на окружающую его среду.

4. Одна из наиболее интересных задач, встречающихся в науках о природе и обществе, — это проблема развития. Еще несколько десятилетий назад подчеркивалось различие в принципах развития систем неживой и живой природы. Развитие замкнутых систем неживой природы — эволюция вещества в изолированных средах — в соответствии со вторым началом термодинамики приводит к увеличению разнообразия, к возрастанию неупорядоченности. Эволюция неживой материи направлена к непрерывной дезорганизации и приводит ее к наиболее вероятному состоянию максимального беспорядка. Напротив, процессы развития живых систем — их эволюция от наименее совершенных биологических макромолекул и микроорганизмов до высокоорганизованных биологических видов — характеризуются в соответствии с эволюционной теорией снижением разнообразия — совершенствованием организации. Ч. Дарвин объяснял необратимую эволюцию к более организованным структурам в живых системах действием естественного отбора, который благоприятствует исключениям и приводит к тому, что исключения — организованные структуры — постепенно становятся правилом. Это объяснение, однако, не устраняет противоречие со вторым началом термодинамики и не снимает вопроса о том, применимы ли законы физики к изучению живых систем.

В последние десятилетия обсуждение этой проблематики продвинулось далеко вперед. Показано, что законы эволю-

ции живого не противоречат второму началу термодинамики. Живой организм — неравновесная открытая система. В открытых системах вдали от состояния равновесия в стационарном режиме уравниения эволюции не исключают спонтанного снижения разнообразия. При определенных условиях возможна самоорганизация материи. Для этого следует «питать» систему свободной энергией, которая используется для управления определенными реакциями, удерживающими систему от увядания до «мертвого» состояния. По-видимому, простота самоорганизации выгодна для живого организма и поэтому эволюция обеспечила самые сложные биологические системы простейшими принципами управления. Эволюция живого отобрала и сохранила лишь те механизмы управления биологическим объектом, которые наиболее экономным образом обеспечивают выполнение функций живого организма

Влияние теории алгоритмической сложности на математические дисциплины

1. Теория алгоритмической сложности всего 20 лет. Тем не менее ее влияние на ряд смежных математических дисциплин (и на их приложения) уже достаточно сильно ощущается.

Теория алгоритмической сложности позволяет более совершенным образом, чем построения, основанные на идеях Р. Мизеса, связать математическую теорию вероятностей с ее приложениями. Приведем рассуждения А. Н. Колмогорова по этому поводу. Традиционно, следуя Р. Мизесу, принимали в качестве основы приложений теории вероятностей признание гипотезы об устойчивости частот. Определим это понятие.

Пусть результат большого числа n испытаний (например, выпадение «орла» и «решки» при подбрасывании не обязательно правильной монеты) записан в виде последовательности нулей и единиц. Будем говорить, что выпадение единицы случайно с вероятностью p , если доля единиц $\frac{m}{n} \sim p$ и эта частота существенно не меняется при выборе из последовательности **любой** достаточно длинной подпоследовательности (при этом выбор должен удовлетворять некоторым нежестким ограничениям, которые мы здесь не оговариваем). Это свойство случайной последовательности,

на котором основывается практическое применение теории вероятностей, называют устойчивостью частот.

Оказывается, что требование устойчивости частот, определяющее допустимость использования теоретико-вероятностных законов, может быть заменено другим, проще проверяемым. Именно алгоритмическая сложность последовательности нулей и единиц, для которой $\frac{m}{n} \sim p$ не может быть существенно больше, чем

$$nH(p) = -n \{p \log p + (1-p) \log (1-p)\}.$$

Можно доказать, что устойчивость частот (по Р. Мизесу) автоматически обеспечивается, если алгоритмическая сложность двоичной последовательности, отражающей результаты испытаний, достаточно близка к $nH(p)$. Этот принцип может быть сформулирован в терминах различных задач теории вероятностей. В частности, пусть последовательность нулей и единиц образует цепь Маркова с матрицей переходных вероятностей $\|p_{ij}\|$, $i, j \in \{0, 1\}$. Здесь p_{00} — приближенное значение частот 0 после 0, p_{01} — частота 1 после 0, p_{10} — частота 0 после 1 и p_{11} — частота 1 после 1. Можно вычислить максимальную алгоритмическую сложность такой последовательности длины n . Оказывается, что если алгоритмическая сложность конкретной последовательности с заданными частотами переходов достаточно близка к максимальной, то для нее автоматически выполняются все утверждения теории цепей Маркова.

Заметим, что в приложениях теории случайных процессов далеко не всегда имеются содержательные аргументы считать процесс марковским. Непосредственная проверка устойчивости частот матрицы переходов по результатам наблюдений — неконструктивная процедура. Оценка сверху алгоритмической сложности последовательности результатов испытаний — существенно более простая задача. Таким образом, теория алгоритмической сложности указывает достаточно конструктивные критерии допустимости использования конкретных экспериментальных статистических материалов для обоснованных предсказаний состояния случайного явления или поведения случайного процесса.

2. Теория алгоритмической сложности оказала революционизирующее влияние на логические основы теории информации. Традиционная теория информации представля-

ет собой раздел теорин вероятностей и как таковая имеет дело с массовыми, а не с индивидуальными явлениями.

Новые принципы построения теории информации не требуют рассмотрения конкретного объекта как реализации некоторого массового явления и не связаны с обработкой статистического материала и использованием вероятностных закономерностей. Таким образом, теория информации, основанная на логических принципах алгоритмической сложности, может предшествовать теории вероятностей, а не опираться на нее.

Шенноновская теория информации и фундаментальные теоремы о передаче информации по своему смыслу применимы только в узкой области, связанной со статистической теорией связи. В теории передачи информации интерес представляют лишь частотные закономерности передаваемых текстов. По словам А. Н. Колмогорова, «вероятностный подход является естественным в теории передачи по каналам связи, передающим «пакеты» информации, состоящие из большого числа не связанных или слабо связанных сообщений, подчиняющихся определенным вероятностным закономерностям» ([22], с. 6).

Поскольку традиционное определение «количества информации» использует только вероятностные концепции, то вопросы типа: «Каково количество информации в одном физическом законе относительно другого закона или в одной математической теореме о другой теореме?» не имеют смысла. Однако такие вопросы вполне естественны и осмысленны при новом алгоритмическом подходе к теории информации. Точно так же корректными становятся вопросы о количестве информации, содержащейся в отдельной книге, в конкретной картине или в определенном организме. Больше того, используя алгоритмическое определение условного количества информации $I(y|x)$ в x об y , можно ставить вопрос о количестве информации, которое получит конкретный объект с фиксированным тезаурусом x от предъявленного ему индивидуального объекта y . И при этом не возникает необходимости вводить ансамбль всех возможных таких объектов и задавать на нем распределение вероятностей. Правда, между содержательной постановкой вопроса и разработкой конструктивных методов подготовки ответа весьма большая дистанция. Здесь должны быть преодолены серьезные трудности. Однако можно ожидать, что эти трудности носят главным образом технический характер. Следует также помнить, что ответы на поставленные

вопросы могут быть получены с точностью до константы, определяемой выбором универсальной «структуры», относительно которой производится оценка. Это означает, что алгоритмическая теория информации применима лишь к большим массивам информации, по отношению к которым константы, лежащие в основе теории, пренебрежимо малы. Тем не менее можно считать, что первый шаг в новом перспективном направлении оценки информации, содержащейся в индивидуальных объектах, сделан. Намечены принципы накопления новых знаний, наиболее экономным образом использующие имеющиеся знания.

3. Утверждения, доказанные в теории алгоритмической сложности, представляют интерес и за пределами теории вероятностей и прикладной теории информации. В частности, ряд результатов теории сложности, по мнению А. Н. Колмогорова, имеет принципиальное значение с точки зрения исследований по основаниям математики. В качестве примера А. Н. Колмогоров приводит одну теорему Барздиня. Утверждение теоремы может быть использовано для некоторых выводов по расширенной постановке X проблемы Гильберта. Этот пример стоит того, чтобы его обсудить.

Напомним предварительно определение понятия «перечислимое множество», используемое в формулировке теоремы Барздиня.

Пусть $Z_+ = \{1, 2, 3, \dots\}$ — множество натуральных чисел, Z_+^m — m -кратное декартово произведение Z_+ на себя.

Множество $M \subset Z_+^m$ называется **перечислимым**, если существует такая частично-рекурсивная функция f , что $M = Df$ (т. е. M является областью определения f). Интуитивный смысл перечислимого множества таков: существует программа, которая распознает элемент $x \in M$, но, возможно, не умеет распознавать элементы $x \in Z_+^m$, $x \notin M$. Другими словами, перечислимое множество — это множество, все элементы которого могут быть получены с помощью некоторой «порождающей» его программы.

Рассмотрим бесконечную двоичную последовательность $x = \{x_1, \dots, x_k, \dots\}$, в которой множество номеров n с $x_n = 1$ перечислимо. Обозначим через $x^n = (x_1, \dots, x_n)$ начальный фрагмент длиной n последовательности x . Если бы было перечислимо и множество номеров нулей, то функция $f(n) = x_n$ была бы вычислимой и сложность $K(x^n | n)$ вычисления по номеру n фрагмента x^n была бы ограниченной. Но в общем случае (при перечислимом мно-

жестве номеров единиц) $K(x^n | n)$ может неограниченно возрасти. Теорема Барздиня дает оценку роста $K(x^n | n)$. Имеет место утверждение: для двоичной последовательности с перечислимым множеством единиц M

$$K(x^n | n) \leq \log n + C,$$

причем существуют такие последовательности, для которых при любом n

$$K(x^n | n) \geq \log n.$$

Напомним теперь X проблему Гильберта и укажем, какие прогнозы можно получить из теоремы Барздиня при расширении этой проблемы.

Д. Гильберт сформулировал свою X проблему следующим образом.

Пусть задано диофантово уравнение с произвольным числом неизвестных и целыми коэффициентами. Требуется указать способ, при помощи которого можно после конечного числа операций установить, разрешимо ли это уравнение в целых числах.

Диофантовы уравнения могут быть закодированы конечными текстами — конечными последовательностями двоичных знаков. Поэтому множество всех диофантовых уравнений может быть перенумеровано натуральными числами. Получим последовательность диофантовых уравнений D_n . В этих терминах X проблема Гильберта может быть переформулирована следующим образом: существует ли общий алгоритм, позволяющий установить, имеет ли уравнение D_n решение в целых числах.

В конце 60-х годов Ю. В. Матиясевич решил эту трудную проблему, поставленную в начале века, и доказал, что такого алгоритма нет. Можно, однако, расширить постановку проблемы и поставить вопрос о существовании алгоритма, позволяющего выяснить, имеют ли целые решения первые n диофантовых уравнений при получении некоторой дополнительной информации J_n с заданным порядком роста количества этой информации с ростом n . Теорема Барздиня показывает, что этот рост может быть очень медленным: $\log n + C$. А. Н. Колмогоров считает, что этот фундаментальный результат — не единственное приложение теоремы Барздиня к основаниям математики.

Предвидимые приложения теории алгоритмической сложности

Как мы видели, теория алгоритмической сложности уже играет заметную роль в математической логике, в теории вероятностей, в теории информации, в основаниях математики. Представляется, однако, что приложения алгоритмической сложности не ограничиваются смежными математическими дисциплинами. Мы уже подчеркивали наше убеждение в том, что на основные утверждения теории алгоритмической сложности можно смотреть как на сформулированные в абстрактных терминах законы природы.

Направление, в котором следует ожидать интерпретации формальных закономерностей теории сложности в содержательных терминах, — это, по-видимому, эволюционная теория в самом общем виде — возникновение и эволюция галактик, возникновение и эволюция Солнечной системы и Земли, химическая эволюция, возникновение жизни и биологическая эволюция, эволюция всей биосферы и отдельных ее видов. Представляется, что больше всех других дисциплин подготовлены к восприятию идей теории сложности биология (точнее, биофизика) и, в частности, генетика. Принципы алгоритмической сложности, по-видимому, со временем скажутся на подходах к проблемам постановки и решения организационных задач в больших коллективах и больших взаимодействующих технических структурах. Информационное содержание структуры — это минимальное число инструкций, необходимых для ее задания. Для задания сложной структуры необходимо много инструкций. Простая повторяющаяся структура может быть задана небольшим числом инструкций. Кардинальная особенность живых организмов и эффективных целенаправленных коллективов — это их высокая организация — сложное взаимодействие, порожденное эволюцией. Уже опубликован ряд работ, в которых идеи алгоритмической сложности используются для определения информационного содержания живого организма. Это пока главным образом работы спекулятивного характера. Некоторые гипотезы по поводу степени организации и активности «живого» содержатся в статье известного специалиста по теории сложности Грегори Дж. Шайтина [42]. В ней содержатся и оригинальные формальные результаты, и список литературы по теме «Сложность и жизнь». Более фундаментальные, но существенно менее формализованные идеи по этой проблематике

содержатся в работах биологов, физиков и главным образом биофизиков Г. Кастлера, М. Эйгена, И. Пригожина, М. В. Волькенштейна, Д. С. Чернавского и др.

В капитальном труде лауреата Нобелевской премии М. Эйгена [46] доказана принципиальная возможность возникновения порядка в открытой системе и снят с обсуждения вопрос о мнимом противоречии биологической эволюции второму началу термодинамики. Миру априори не грозит тепловая смерть. Эволюция живой материи в полном соответствии с законами физики и химии способна обеспечить возникновение новой информации, совершенствование организации. Исходный материал для эволюции — случайные ненаправленные мутации. Управляющие факторы эволюции — естественный отбор и сложившийся к моменту отбора тип структуры и развития системы (организма). «Технологической базой» эволюции живой природы на всех уровнях, начиная с молекулярного, является стандартизация — комбинирование новых структур из ограниченного числа элементов, сформировавшихся на предбиологической стадии эволюции. При другой технологии, связанной, например, с перебором всевозможных мутаций, вероятность возникновения жизни за время существования Вселенной была бы ничтожно мала.

Математическая модель эволюции на том или ином уровне представляет собой систему обычно нелинейных дифференциальных уравнений (обыкновенных или в частных производных в зависимости от того, учитывается ли влияние пространственных координат на скорость изменения и другие параметры взаимодействия элементов системы). Переменные кинетических уравнений в соответствии с рассматриваемым уровнем эволюции представляют собой концентрации реагирующих веществ, число микроорганизмов или их суммарную биомассу, численность вида. Уравнения эволюции связывают скорости изменения переменных, описывающих развитие системы, с характеристиками взаимодействия ее элементов.

В последние годы в биофизике исследуются различные варианты моделей эволюции — кинетические уравнения, определяющие траектории эволюционного процесса, изучаются химические, биологические, энергетические и информационные аспекты самоорганизации материи. Основным вывод, к которому приводит качественный анализ уравнений эволюционного процесса, заключается в том, что возникновение жизни и последующая биологическая эволю-

ция — явления возрастания упорядоченности открытой, далекой от равновесия диссипативной системы. Представляется, что удачная формализация этого тезиса, от которой следует ожидать не только качественных, но и количественных оценок эволюционного процесса, может быть получена в терминах алгоритмической сложности.

Пусть $y(t)$ — траектория эволюционного процесса. Она зависит от начальных условий³, от экологической ситуации, в которой протекал процесс, и от реализации управляющего фактора — естественного отбора. Прогноз эволюционного процесса на этапе t требует знания истории, т. е. последовательности $y^{t-1} = \{y(0), y(1), \dots, y(t-1)\}$, и стимула, определяющего ход развития.

Неживая материя в соответствии со вторым началом термодинамики эволюционирует к состоянию максимальной неопределенности — к термодинамическому равновесию. Эволюция живой материи — открытой системы, далекой от равновесия, — стимулируется стремлением к накоплению полезной (ценной) информации, способствующей повышению упорядоченности системы.

Под мерой ценности информации здесь естественно подразумевать величину $I(y(t)|y^{t-1}) = I(y(t)|y(0), y(1), \dots, y(t-1))$, которая в комбинаторной трактовке теоретико-информационных понятий вычисляется как разность между сложностью системы на этапе t и ее условной сложностью на этом этапе, определяемой фрагментом y^{t-1} эволюционного процесса. Чем больше ценность информации, тем лучше организована, более упорядочена система на этапе t . Это значит, что естественный ход эволюции приводит к росту

$$I(y(t)|y(0), \dots, y(t-1)) = K(y(t)) - K(y(t)|y^{t-1}).$$

Условия протекания эволюции не произвольны, а связаны определенными ограничениями, в частности ограничениями экологического характера. Установление этих ограничений дает основания для построения мажоранты сложности $K(y(t))$ и условной сложности $K(y(t)|y^{t-1})$. Таким образом, может быть описана область, в которой организация (упорядоченность) системы возрастает. При этом возникает и принципиальная возможность учета антропологического вмешательства в естественный ход эволюции.

³ Если бы жизнь возникла и развивалась вновь, ход эволюции был бы, по-видимому, иным.

ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ

Меры сложности вычислений

1. Алгоритмическая сложность — предмет рассмотрения предыдущей главы ($K(y, x)$ и $K(x)$) — введена для оценки сложности решения индивидуальных задач и сложности описания индивидуальных объектов. В теории вычислений существенный интерес представляют характеристики сложности массовых задач — классов X «однотипных» индивидуальных задач x .

Зафиксируем способ кодирования индивидуальных задач и способ кодирования их решений. Можно считать соответствующие коды элементами множества целых неотрицательных чисел $Z = \{0, 1, 2, \dots\}$. Тогда каждой массовой задаче соответствует функция $f: Z \rightarrow Z$. Функция $f(x)$ — это код y решения индивидуальной задачи, код условий которой x . Будем говорить, что массовая задача разрешима на машине со структурой $\varphi(x, p)$, если найдется программа p , для которой $\varphi(x, p) = f(x) \quad \forall x \in Z$. Иначе говоря, класс задач разрешим, если существует программа, позволяющая найти решение любой задачи класса по предъявленному ее условию. Подобных программ, разумеется, может быть много. Минимальную из длин таких программ естественно называть статической алгоритмической сложностью соответствующего класса задач. Будем обозначать эту характеристику $K_\varphi(f)$. Таким образом,

$$K_\varphi(f) = \begin{cases} \min l(p) \mid p: \varphi(x, p) = f(x), \\ \infty: \forall p \in S \quad \exists x \in X \quad \varphi(x, p) \neq f(x). \end{cases}$$

Здесь по-прежнему S — множество всех допустимых программ. (Если кодирование задач и решений проводится в двоичных последовательностях, то S — множество всех двоичных последовательностей.) Как и для алгоритмической сложности индивидуального объекта, можно показать, что существуют универсальные (оптимальные) структуры ψ , такие, что любая задача, разрешимая на машине любой структуры φ , разрешима и на машине структуры ψ , причем $K_\psi(f) \leq K_\varphi(f) + C_{\psi\varphi}$, где $C_{\psi\varphi}$ — константа, зависящая только от структур ψ и φ .

Массовые задачи, разрешимые на машине универсальной структуры (это в точности задачи, которые отвечают вычислимым функциям f), называются алгоритмически раз-

решимыми. Соответственно $K_\psi(f)$ (для фиксированной оптимальной структуры ψ) будем называть статической алгоритмической сложностью массовой задачи класса задач f и обозначать $K(f)$.

Таким образом, алгоритмическая сложность массовой задачи — это сложность описания алгоритма решения задач данного класса. Она оценивает минимально возможную длину программы решения фиксированной массовой проблемы, но не дает представления о динамических (внешних) характеристиках вычислительного процесса. Алгоритмическая сложность класса задач не позволяет судить о трудоемкости вычислений при решении любой индивидуальной задачи и оценивать средние или максимальные затраты времени, памяти и других ресурсов, необходимых для ее решения. А ведь именно эти вопросы главным образом и представляют интерес для вычислителя. Часто оказывается, что сокращение длины программы приводит к существенному увеличению времени счета и потребной памяти. Известны задачи (например, так называемые переборные задачи), для которых существуют весьма простые алгоритмы с короткими программами, однако получение решения с помощью этих простых программ требует астрономического числа вычислений. Встречаются и алгоритмы с громоздкими программами, которые тем не менее сравнительно быстро приводят к цели. Таким образом, для более полной оценки вычислительного процесса следует дополнить внутреннюю характеристику трудоемкости процесса — алгоритмическую сложность — внешними, позволяющими оценить минимальные затраты различных вычислительных ресурсов на решение задач класса.

Чтобы проиллюстрировать, к чему может привести недооценка всесторонней характеристики вычислительного процесса, приведем следующий (сообщенный нам Б. Т. Поляком) курьезный случай.

В 1972 г. в № 2 журнала «Математическое программирование» появилась за подписью Anonimus статья «Новый алгоритм оптимизации». Статья написана на сугубо формальном языке, свойственном многим абстрактным математическим работам. За частоклоком определений, лемм, теорем трудно было разобраться в содержательной структуре метода, а приведенная на некоем фантастическом языке программа вычислений не способствовала уяснению принципов, на которых основан предлагаемый метод. В библиографии, на которую ссылался автор, фигурировали

Гаусс, Бурбаки, неопубликованная статья автора и частное сообщение.

Для демонстрации достоинств «нового» метода и его преимуществ по сравнению с известными алгоритмами автор приводит таблицу, в которой различные методы сравниваются по 10 критериям. Среди них применимость метода к дискретным задачам, к оптимизации негладких и невыпуклых функций, длина программы (в наших терминах — алгоритмическая сложность), наличие доказательства сходимости и др. Показано, что по 9 из 10 критериев «новый» метод имеет решающее преимущество по сравнению с известными. Лишь по последней характеристике — по вычислительным ресурсам, потребным для решения задач (в нашей терминологии — по компонентам вычислительной сложности), в работе отсутствуют данные как по «новому» методу, так и по существующим. Тем не менее явное превосходство «нового» метода по 9 из 10 критериев оценки дало автору основание рекомендовать предлагаемую им процедуру как «универсальный метод оптимизации», который следует предпочесть ранее разработанным.

Как оказалось, за нарочито чрезмерно формализованным абстрактным описанием «нового» метода скрывался обычный полный перебор, а автором статьи-шутки был известный математик Филипп Вулф. Отношение массового потребителя вычислительных алгоритмов к аргументации статьи характеризует то, что, по сообщению Ф. Вулфа, многие читатели восприняли эту пародию всерьез.

Таким образом, алгоритмическая сложность и вычислительная сложность характеризуют различные аспекты вычислительных процедур. Объективная оценка трудоемкости вычислений должна учитывать различные компоненты сложности.

Статическая сложность решения задач фиксированного класса — скалярная величина. Сложность вычислительного процесса — функция, аргумент которой определяется индивидуальной задачей класса. Различные алгоритмы всегда сравнимы между собой по длине («объему») программы, а процессы вычислений могут быть несравнимы по сложности. Для разных задач класса (для разных значений аргумента вычислительной сложности) те или иные процессы вычислений могут оказаться более экономными. Для алгоритмов, таким образом, может быть установлена сложностная иерархия, а характеристики классов задач, основанные на мерах сложности вычислений, лишь частично

упорядочены. Для каждой вычислимой функции (для каждого класса задач, или, что то же, для каждой массовой проблемы) можно упорядочить по минимальному объему программы все алгоритмы, позволяющие решать задачи этого класса. С упорядочиванием алгоритмов по вычислительной сложности ситуация менее прозрачна и возможные утверждения более расплывчаты. Однако об этом несколько позже.

Для оценки эффективности и трудоемкости вычислительных процессов в зависимости от ситуации используются статистические или динамические меры сложности. В тех случаях, когда время программирования и трудоемкость программирования велики или когда программа используется редко, подходящей характеристикой процесса вычислений является объем программы. В случаях, когда программа используется часто или потребности в ресурсах ЭВМ велики, целесообразно оценивать вычислительный процесс составляющими его вычислительной сложности.

Рассмотрим пару (A, x) , где A — алгоритм решения массовой проблемы (класса задач), а x — индивидуальная задача класса. Вычислительный процесс характеризуется функциями сложности $\mu_A(x)$, называемыми **сигнализирующими вычисления** [38]. Чтобы упростить сравнение вычислительных процессов, целесообразно характеризовать решение массовых проблем верхними и нижними оценками сигнализирующих. Получить верхнюю оценку $\bar{\mu}$ сигнализирующей $\mu_A(x)$, значит, показать, что существует алгоритм A , решающий массовую проблему $f(x)$ с характеристикой сложности, не превосходящей $\bar{\mu}$. Сигнализирующая $\mu_A(x)$ имеет нижнюю оценку $\underline{\mu}$, если для любого алгоритма A , решающего задачи x класса $f(x)$, сигнализирующая $\mu_A(x)$ не меньше $\underline{\mu}$. Чем ближе друг к другу $\bar{\mu}$ и $\underline{\mu}$, тем точнее охарактеризована соответствующая компонента вычислительной сложности. Рассматриваются также «средние» и гарантированные оценки сигнализирующих и зависимости этих оценок от размерности задачи.

2. В теории вычислений рассматриваются следующие меры сложности.

(1) **Сигнализирующая времени** $t_A(x)$, равная числу шагов, которые совершает алгоритм A , будучи применен к слову x (число шагов, необходимых алгоритму A для решения индивидуальной задачи x). Иногда используются также

$$t_A(X) = \max_{x \in X} t_A(x) \text{ и } t_A(n) = \max_{|x|=n} t_A(x),$$

где $|x|$ — длина слова x (или размерность задачи x , принадлежащей данному классу X). Меры сложности вычислений $t_A(X)$ и $t_A(n)$ относятся к худшему случаю и дают гарантированные оценки затрат времени на решение задачи фиксированного класса и фиксированной размерности. Гарантированные оценки сложности массовой проблемы часто оказываются чрезмерно завышенными. Задачи x , на которых они достигаются, могут встречаться крайне редко. Например, для симплексного метода в линейном программировании построены искусственные примеры, в которых временная сложность экспоненциально растет с ростом размерности задачи. Между тем на практике симплекс-метод оказывается весьма эффективным. Более реалистичны оценки временной сложности «в среднем». Однако далеко не всегда имеются основания определить частоту возникновения тех или иных задач данного класса.

Анализ вычислительной сложности требует также оценки таких временных сигнализирующих, как $t(X) = \min_A t_A(X)$ и $t(n) = \min_A t_A(n)$, где в зависимости от исследуемой проблемы минимизация производится по всевозможным алгоритмам или по алгоритмам некоторого фиксированного класса.

Сигнализирующая времени $t_A(x)$ характеризует время работы алгоритма A , затрачиваемое на решение задачи x (от момента записи входной информации до получения результата).

(2) Сигнализирующая емкости $S_A(x)$, равная числу ячеек памяти, к которым хотя бы раз происходило обращение в процессе вычисления слова x (решения задачи x) с помощью алгоритма A . Аналогично вводятся $S_A(X) = \max_{x \in X} S_A(x)$ и $S_A(n) = \max_{|x|=n} S_A(x)$. Все рассуждения об оценках «по наихудшему случаю» и «в среднем», приведенные по отношению к сигнализирующей времени, переносятся, естественно, и на сигнализирующую емкости. Сигнализирующая емкости характеризует объем памяти, необходимый для решения задачи.

(3) Сигнализирующая колебаний (поворотов) $\omega_A(x)$ определяется для машин Тьюринга, вычисляющих слово x в

соответствии с алгоритмом A как число изменений направления при движении считывающей головки вдоль ленты. Аналогом сигнализирующей $\omega_A(x)$ является количество циклов в программе для реальной машины, которые нарушают «естественный» ход программы.

(4) **Сигнализирующая режима** $r_A(x)$ определяется для машины Тьюринга как число переходов из ячейки в ячейку, которые совершает алгоритм A , будучи применен к слову x . Сигнализирующая $r_A(x)$ учитывает только те рабочие такты, при которых считывается новая информация. Для реальных адресных машин выполнение каждой команды сопровождается обращением к памяти. Поэтому аналогом сигнализирующей режима является число обращений к долговременным запоминающим устройствам-накопителям на магнитных лентах, дисках, барабанах и т. д.

В теории вычислений рассматриваются и другие меры сложности, удовлетворяющие некоторым естественным условиям — требованиям к характеристикам трудоемкости вычисления. Однако наибольшее внимание в известных работах уделяется временной и емкостной сигнализирующим. Временная сигнализирующая часто отождествляется с вычислительной сложностью.

Для рациональной организации вычислений целесообразно учитывать и меры сложности, характеризующие надежность вычислений, возможности распараллеливания процесса решения задач класса, частоту обращения к различным ячейкам и блокам машины. Построение обобщенных и в то же время достаточно близких к реальным моделей вычислительного устройства и выбор сигнализирующих устройств и систем математического обеспечения — важная задача прикладной теории сложности.

Инвариантные результаты теории вычислений

1. Выше уже указывалось, что сигнализирующие функции могут оказаться не сравнимыми между собой. Поэтому, если задана некоторая вычислимая функция, то неясно, существует ли для нее наилучшее (по той или иной мере сложности) вычисление. Сравнение классов задач (вычислимых функций) по вычислительной сложности приходится вести по верхним и нижним оценкам сложности или по так или иначе определенным средним (по задачам класса) значениям сигнализирующих.

Выводы теории вычислительной сложности зависят от выбранного вычислительного устройства и выбранной меры вычислительной сложности. Поэтому можно предположить, что переход от одного типа вычислителя к другому и от одной сигнализирующей к другой изменит и сложностную иерархию функций (например, по верхним или нижним оценкам $\bar{\mu}$ или $\underline{\mu}$). В некотором смысле это так и есть на самом деле. Однако в аксиоматической теории вычислений установлены весьма общие инвариантные результаты. Суть их в том, что выбор сигнализирующих вычисления и типа вычислителя не может существенно изменить соотношения между сложностями различных классов задач. Если задача I значительно сложнее задачи II, то это отношение сохранится при любой естественной сигнализирующей, выбранной в качестве меры сложности, и любом типе вычислителя, решающего эти задачи.

2. Общие закономерности процессов вычисления проявляются лишь при рассмотрении достаточно богатых по своим возможностям классов объектов. Если интересоваться, например, такой «узкой» массовой проблемой, как перемножение n -разрядных чисел, окажется, что временные сигнализирующие различаются по порядку величины для различных типов вычислительных устройств. Аксиоматическая теория вычислений рассматривает гораздо более широкие классы задач и находит в них объекты, для которых справедливы весьма общие, не зависящие от организации процесса вычисления или устройства вычислителя закономерности.

Приведем пример такого общего результата. Пусть мы имеем две универсальные вычислительные машины. При этом первая из них обладает богатым множеством операций, гибкой структурой и огромным быстродействием, а вторая во всех отношениях уступает первой — допускает всего несколько операций, имеет жесткую структуру и весьма низкую производительность. Оказывается, существуют вычислимы функции, для вычисления которых нельзя получить преимущество в скорости вычислений, используя первую машину. Для каждой программы мощной машины существует программа малопроизводительной машины, вычисляющая ту же функцию быстрее (при достаточно больших значениях аргумента). Однако не существует эффективного способа нахождения такой программы для второй машины. Доказано лишь, что она существует. Аналогичные результаты можно получить и для любой другой сиг-

нализирующей вычислений, удовлетворяющей естественным требованиям к мерам сложности.

Заметим, что подобные результаты не зависят от структуры используемых машин. Как уже указывалось, наиболее существенная информация о вычислительной сложности — это информация, не зависящая от типа машины. Хорошие алгоритмы имеют тенденцию оставаться хорошими независимо от языка программирования или типа машины, на которой они реализованы.

Отношение к таким результатам может быть двоякое. С одной стороны, они позволяют описать структуру и закономерности класса вычислимых функций и весьма привлекательны своей законченностью. С другой — эта теория рассматривает задачи, слишком сложные с точки зрения сегодняшней практики. А как утверждает Л. Заде [15], чем сложнее система, тем менее мы способны дать точные и в то же время имеющие практическое значение суждения о ее поведении. Это вечное противоречие между возможностями теории и требованиями практики стимулирует возникновение новых идей и разработку новых подходов к постановке и решению актуальных задач современности.

Классификация по сложности задач дискретной оптимизации

1. Многие задачи планирования в экономике и проектирования в технике, будучи формализованы, могут быть записаны как задачи дискретной оптимизации (целочисленного программирования). В формальных терминах решение такой задачи — это вектор параметров управления x с дискретными компонентами (каждая из которых может принимать одно из нескольких фиксированных значений). Решение задачи оптимизирует заданную функцию $f_0(x)$ — показатель качества плана или проекта — при соблюдении ограничений вида $f_j(x) \leq 0$, $j=1, \dots, m$. При линейных функциях, определяющих показатель качества и ограничения, — это задача дискретного линейного программирования.

В терминах задач дискретного (целочисленного) линейного программирования описываются такие экономические и технические проблемы, как задачи транспортного типа, задачи назначения, распределительные задачи, задачи теории расписаний, задача коммивояжера, задачи логи-

ческого, технологического и надежностного проектирования, и многие другие задачи комбинаторного типа. Содержательному описанию задач, интерпретации и численным методам дискретной оптимизации посвящена богатая литература (см., например, [55] и цитированные в ней работы).

Практика решения задач целочисленного программирования выявила некоторые классы задач, решение которых не связано с трудоемкими вычислениями, и многие задачи, для которых до сих пор не найдены экономные методы решения. Уже давно — лет двадцать назад — было замечено, что при решении ряда задач дискретной оптимизации любым из известных методов требуется чрезмерно большое число итераций (шагов) и трудоемкость решения быстро растет с ростом размерности задачи. Богатый опыт успешного решения задач линейного программирования большой размерности, в которых отсутствовало требование целочисленности переменных, породил мнение о том, что трудности, возникавшие при решении отдельных дискретных задач, — явление временное, связанное с несовершенством разработанных методов дискретной оптимизации и недостаточной производительностью вычислительной техники тех лет. Ситуация, однако, оказалась гораздо более сложной, чем это предполагалось ранее.

2. В начале семидесятых годов появились работы, которые дали основание для вывода, что трудоемкость решения ряда задач комбинаторного типа (впрочем, как и задач организации многих других вычислительных процессов) определяется не преходящими факторами, связанными с несовершенством вычислительных методов и невысокой производительностью ЭВМ, а вызвана, по-видимому, принципиальными особенностями структуры этих задач.

В задачах линейного программирования, в которых переменные не ограничены условиями целочисленности, глобальная информация об условиях задачи полностью определяется локальной информацией о значениях функционалов качества и ограничений и их градиентов в любой точке области задания переменных. Этим в известной мере определяется относительная простота и эффективность методов линейного программирования. В задачах целочисленного линейного программирования ситуация иная. Пусть, например, все n переменных задачи булевы, т. е. могут принимать только значения 0 или 1. Это значит, что область определения задачи — множество вершин n -мерного куба. При отсутствии априорных данных о структуре

целевого функционала и матрицы условий локальная информация о значениях функционалов задачи, достигаемых в какой-либо вершине области определения решения, не дает сведений о значениях, достигаемых этими функционалами в соседних вершинах куба. По-видимому, в этом одна из главных причин трудоемкости решения многих задач дискретной оптимизации. Более строгие, однако еще не окончательные суждения по этому поводу (полностью обоснованных выводов еще нет) будут изложены далее.

3. Мы говорили о легкорешаемых и труднорешаемых задачах дискретной оптимизации. Как же определить степень трудоемкости вычислительного процесса?

По предложению Дж. Эдмондса [45] массовая задача считается легкорешаемой, если существует алгоритм, позволяющий решать любую индивидуальную задачу, принадлежащую этой массовой проблеме, за число шагов (или при числе элементарных операций), не превышающее некоторый полином от размерности задачи. Соответственно, массовая задача трудноразрешима, если не существует алгоритма, позволяющего решать все задачи этого класса при числе итераций, не меньшем, чем экспоненциальная функция от размерности задачи. Таким образом, «хорошие задачи» имеют полиномиальную временную сложность, а сложность «плохих задач» экспоненциальна.

Следующие три таблицы (заимствованные из [12] и [37]) дают наглядное представление о сравнительной скорости роста некоторых функций полиномиального и экспоненци-

Таблица 1

Оценка времени решения задачи

Размер- ность n	20	50	100	200	500	1000
Сложность						
1000 n	0,02 с	0,05 с	0,1 с	0,2 с	0,5 с	1 с
1000 $n \log n$	0,09 с	0,3 с	0,6 с	1,5 с	4,5 с	10 с
100 n^2	0,04 с	0,25 с	1 с	4 с	25 с	2 мин
10 n^3	0,02 с	1 с	10 с	1 мин	21 мин	2,7 ч
$n^{\log n}$	0,4 с	1,1 ч	220 сут	125 век	$5 \cdot 10^8$ век	
$2^{n/3}$	0,0001 с	0,1 с	2,7 ч	$3 \cdot 10^4$ век		
2^n	1 с	35 лет	$3 \cdot 10^4$ век			
3^n	58 мин	$2 \cdot 10^9$ век				

**Влияние производительности ЭВМ
на размерность разрешимой задачи**

Производи- тельность Сложность	Современные ЭВМ	ЭВМ, в 100 раз более быстрые	ЭВМ, в 1000 раз более быстрые
n	N_1	100 N_1	1000 N_1
n^2	N_2	10 N_2	31,6 N_2
n^3	N_3	4,64 N_3	10 N_3
n^5	N_4	2,5 N_4	4 N_4
2^n	N_5	$N_5 + 6,64$	$N_5 + 10$
3^n	N_6	$N_6 + 4,19$	$N_6 + 6,3$

ального характера и, следовательно, позволяют сравнить значения вычислительных сложностей (временных сигнализирующих), отвечающих «хорошим» и «плохим» задачам. Правда, необходимо иметь в виду, что при относительно малых n экспоненциальный алгоритм может оказаться эффективнее алгоритма с полиномиально ограниченным временем работы. Например, функция 2^n не превосходит n^2 до значения n , равного 4, не превосходит n^3 до $n=9$, не превосходит n^5 до $n=20$, не превышает n^{10} до $n=59$ и т. д. Тем не менее, как видно из приведенных таблиц, для задач немалой размерности, решение которых, собственно, и составляет проблему дискретной оптимизации, разделение задач по сложности на полиномиально и экспоненциально разрешимые естественно и обоснованно. Трудноразрешимые

Максимальная размерность задач,

Время Сложность	1 с	10^2 с (1,7 мин)
1000 n	10^3	10^5
1000 $n \log n$	$1,4 \cdot 10^2$	$7,7 \cdot 10^3$
100 n^2	10^2	10^3
$10n^3$	46	$2,1 \cdot 10^2$
$n^{\log n}$	22	36
$2^{n/3}$	59	79
2^n	19	26
3^n	12	16

задачи — это, грубо говоря, задачи, требующие для решения почти полного перебора вариантов. Как видно из таблиц, при немалых n ($n=50$ и более) вычисления экспоненциальной временной сложности практически нереализуемы. Существенный рост производительности ЭВМ не приведет к заметному увеличению размерности разрешимых задач с экспоненциальной временной сигнализирующей.

Таблица 1 позволяет сравнить скорости роста нескольких типичных полиномиальных и экспоненциальных функций. Таблица оценивает время решения задач размерности n при разных временных сигнализирующих. Предполагается, что один шаг соответствует 1 мкс. Из таблицы видно, что по мере роста размерности n влияние постоянного множителя при степенных функциях на время счета убывает.

Таблица 2 показывает, как увеличатся размерности задач разной сложности, решаемых за 1 ч машинного времени, если производительность ЭВМ возрастет в 100 или в 1000 раз по сравнению с современными машинами.

Таблица 3, характеризующая максимальную размерность задач, разрешимых за данное время при различных временных сигнализирующих, иллюстрирует с несколько иной точки зрения выводы, определяемые таблицей 2.

Приведенные таблицы обосновывают целесообразность предложенной Дж. Эдмондсом классификации по сложности комбинаторных задач. Такая классификация, как мы увидим дальше, обладает и рядом других достоинств. Она облегчает оценку сложности одних массовых задач по оценкам сложности других задач. Кроме того, она слабо чув-

Т а б л и ц а 3

разрешимых за данное время

10^4 с (2,7 ч)	10^6 с (12 сут)	10^8 с (3 года)	10^{10} с (3 века)
10^7	10^9	10^{11}	10^{13}
$5,2 \cdot 10^5$	$3,9 \cdot 10^7$	$3,1 \cdot 10^9$	$2,6 \cdot 10^{11}$
10^4	10^5	10^8	10^7
10^3	$4,6 \cdot 10^3$	$2,1 \cdot 10^4$	10^5
54	79	112	156
99	119	139	159
33	39	46	53
20	25	29	33

ствительна к структуре машин, на которых производятся вычисления.

Большинство известных полиномиальных задач имеет полиномиальную оценку временной сложности невысокой степени ($O(n^3)$ или лучше). Выяснение, какие из известных естественных комбинаторных проблем являются полиномиальными, а какие экспоненциальными — одна из важнейших задач теории вычислительной сложности.

В теории вычислительной сложности основное внимание уделяется временной сигнализирующей. Однако известны и некоторые работы по сигнализирующей емкости. В частности, из результатов Хартманиса и Стирнса следует, что имеются проблемы, разрешимые при экспоненциальной памяти, но не разрешимые при полиномиальной памяти.

4. Рост производительности вычислительной техники и возможность реализации на современных машинах все более сложных алгоритмов повышают интерес практиков к теории сложности.

Следует, однако, иметь в виду, что ряд общих результатов теории сложности получен ценой некоторых практически нереализуемых предположений. В частности, понятие алгоритмической сложности не предполагает априори никаких ограничений на время работы алгоритма. Если наложить даже слабые ограничения на число шагов алгоритма, минимальная длина программы может существенно возрасти. Как уже отмечалось, колмогоровская сложность объекта не является вычислимой функцией. Оптимальный алгоритм будет перебирать все возможные способы кодирования объекта. Если время работы алгоритма заранее ограничено, не будет уверенности в том, что лучший из рассмотренных к этому времени способов кодировки определит минимальную длину программы, восстанавливающей объект.

Аналогичным образом обстоит дело и в теории вычислительной сложности, которая игнорирует вопрос об используемых ресурсах. Это ограничивает способность теории определять минимальные затраты вычислительных ресурсов в реальных ситуациях. Учет ограничений на организацию вычислительного процесса необходим не только для получения более «реальных» оценок сложности, но и для обоснованного решения практических задач, связанных с проектированием ЭВМ, в частности, для определения возможности размена памяти на время в процессе вычислений. Работы последних лет по теории сложности уделяют все

больше внимания выяснению зависимости между качеством решения задачи и затрачиваемыми при этом ресурсами.

Заметим, что практически все, что было сказано здесь о разделении задач дискретной оптимизации на легко разрешимые и трудноразрешимые, можно перенести и на другие массовые задачи организации вычислений — на задачи распознавания свойств, на установление значений вычисляемых функций в заданной области изменения дискретных аргументов и др.

Легкорешаемые дискретные задачи

1. Известно относительно небольшое количество массовых комбинаторных задач, для которых разработаны простые (полиномиальные) алгоритмы точного решения. Значительно больше задач, для которых имеются полиномиальные приближенные методы решения, гарантирующие заданную верхнюю границу допустимых погрешностей. Известны и такие задачи дискретной оптимизации (их совсем немного), для которых требование дискретности параметров управления не вносит никаких дополнительных усложнений по сравнению с соответствующими непрерывными задачами.

Рассмотрим, например, традиционную задачу линейного программирования. Требуется вычислить n -мерный вектор $x = (x_1, \dots, x_n)$, минимизирующий линейную форму $\sum_{j=1}^n c_j x_j$ при соблюдении линейных ограничений $\sum_{j=1}^n a_{ij} x_j = b_i, i = 1,$

$\dots, m, x_j \geq 0, j = 1, \dots, n$. Будем считать элементы a_{ij} матрицы условий $A = \|a_{ij}\|$ фиксированными целыми числами, а компоненты b_i вектора b — произвольными целочисленными величинами. Пусть, кроме того, ранг матрицы A совпадает с числом m ограничений-равенств. Как известно, область определения задачи линейного программирования представляет собой выпуклое многогранное множество, обусловленное ограничениями задачи. Обозначим это множество через $M(b)$. Оптимальное значение целевого функционала задачи линейного программирования (если оно единственное) достигается в некоторой вершине $M(b)$ (если решение не единственное, то среди множества решений задачи имеется несколько (по крайней мере две) вершин $M(b)$, а все остальные

решения являются выпуклыми комбинациями этих вершин). Следовательно, если матрица A устроена так, что все вершины многогранного множества $M(b)$ имеют целочисленные координаты, то дополнительное требование целочисленности искомых переменных не внесет никаких изменений в алгоритм решения соответствующей непрерывной задачи и не скажется на трудоемкости использования этого алгоритма. Что же касается задач линейного программирования, в которых не требуется гарантировать целочисленность искомых переменных, то для них имеются полиномиальные методы решения (подробней об этом далее).

Оказывается, существуют такие целочисленные матрицы A , которые гарантируют целочисленность всех вершин множества $M(b)$ при произвольных целых значениях компонент вектора b .

Имеет место утверждение.

Для целочисленности всех вершин многогранных множеств $M(b)$ с произвольными целочисленными векторами b необходимо и достаточно, чтобы любой минор порядка m матрицы условий A был равен 0 или ± 1 .

Таким образом, задачи целочисленного линейного программирования, матрицы ограничений которых удовлетворяют условиям приведенного утверждения, представляют собой в принятой терминологии класс легкорешаемых задач.

Сформулированным условиям удовлетворяют, в частности, классические транспортные задачи. Жесткость условий утверждения означает, что класс задач линейного программирования, в которых дополнительное требование дискретности искомых переменных не усложняет методов решения, крайне беден.

Строго говоря, назвать классическую транспортную задачу легкорешаемой стало возможным лишь после 1979 г., когда было показано, что для решения общей задачи линейного программирования существует полиномиальный алгоритм. Дело в том, что стандартные методы решения транспортной задачи представляют собой реализации общих методов линейного программирования, учитывающие специфику матрицы условий. Так, например, метод потенциалов представляет собой симплекс-метод, приспособленный к условиям транспортной задачи, а известный венгерский метод является реализацией общего метода сокращения невязок. Уже указывалось, что все ранее известные общие конечные методы линейного программирования не обладают

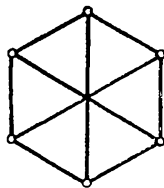
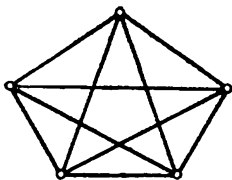
полиномиальной сходимостью. Для каждого из них построены примеры, решение которых требует перебора почти всех вершин многогранного множества условий задачи. Больше того, были найдены и примеры транспортных задач размерности n ($n = \min(n_1, n_2)$), где n_1 — число пунктов производства, а n_2 — число пунктов потребления) для решения которых число итераций экспоненциально растет с ростом n . Полиномиальный алгоритм, построенный в 1979 г. Л. Г. Хачияном [40] для решения общей задачи линейного программирования, может быть переформулирован с учетом особенностей матрицы условий транспортной задачи в терминах этой задачи, и это дает основание считать классическую транспортную задачу легко решаемой.

2. Одна из наиболее часто встречающихся вычислительных операций, представляющих как самостоятельный интерес, так и служащих вспомогательной процедурой для решения более сложных задач, — это вычисление произведения двух матриц порядка n . Стандартные методы умножения матриц требуют $O(n^3)$ элементарных операций. Штрассен построил алгоритм умножения матриц, для которого необходимо только $O(n^{\log_2 7})$ операций ($\log_2 7 = 2,807$). Этот алгоритм был улучшен в 1979 г. Новый алгоритм гарантирует умножение матриц за $O\left(n^{\frac{3 \log 17}{\log 26}}\right)$ операций ($\frac{3 \log 17}{\log 26} = 2,609$). Недавно появилась информация, относящаяся к 1981 г., об оценке трудоемкости перемножения матриц величиной порядка $O(n^{2,4915})$.

Так называемая задача о кратчайших путях — задача нахождения минимальных расстояний от данной вершины ориентированного графа до всех остальных вершин — связана с умножением матриц. Дейкстра предложил алгоритм решения задачи о кратчайших путях, требующий в зависимости от реализации графа $O(n^2)$ или $O(m \ln n)$ операций, где n — число вершин, а m — число ребер графа.

Нахождение кратчайших путей для всех пар вершин графа требует $O(n^3)$ операций (Фloyd). Фредман построил более экономный алгоритм решения этой задачи, использующий $O\left(n^3 \left(\frac{\log \log n}{\log n}\right)^{1/3}\right)$ элементарных операций и среди них $O(n^{2,5})$ сравнений.

Как видим, оценка сверху трудоемкости решения массовых вычислительных задач требует «изобретения» алгоритма, решающего все задачи класса, и установления



зависимости вычислительной сложности от размерности задачи.

3. В теории обработки сигналов и в различных задачах вычислительной математики (например, при построении экономных алгоритмов вычисления полиномов и умножения полиномов) в качестве вспомогательной задачи используется дискретное преобразование Фурье. Напомним, что дискретным преобразованием Фурье n -мерного вектора $a = (a_0, a_1, \dots, a_{n-1})$ называется вектор $b = (b_0, b_1, \dots, b_{n-1})$ такой, что $b_k = \sum_{i=0}^{n-1} a_i \omega_i^k$, где $\omega_0, \omega_1, \dots, \omega_{n-1}$ — (комплексные) корни n -ой степени из единицы. Традиционный метод вычисления дискретного преобразования Фурье требует $O(n^2)$ операций. Так называемое быстрое преобразование Фурье производится за $O(n \log n)$ операций. Последний результат в этом направлении получен Виноградом, который предложил алгоритм вычисления дискретного преобразования Фурье за $O(n)$ умножений. Однако оценки числа других элементарных операций, потребных при этом, Виноград не приводит.

4. При автоматизации проектирования радиоэлектронной аппаратуры встречается следующая задача. На плате расположено некоторое число элементов, которые должны быть соединены между собой в соответствии с заданной схемой. Возникает вопрос: можно ли реализовать эту схему на плоскости без самопересечений. Другими словами, можно ли нарисовать заданный граф на плоскости так, чтобы никакие два ребра не пересекались. Эта задача называется проверкой планарности графов. Куратовский показал в 1930 г., что граф планарен в том и только в том случае, если он не содержит подграфов, гомеоморфных указанным на рисунке двум графам K_5 и $K_{3,3}$. По сообщению В. Б. Алексеева, переведившего обзор Р. Э. Тарьяна [37], содержащий некоторые из приведенных здесь оценок, этот же результат доказан, но не опубликован Л. С. Понтрягиным в 1927 г.

Шпир построил алгоритм проверки планарности графа, требующий $O(n^3)$ операций. Хопкрофт и Тарьян опубликовали более экономный алгоритм, позволяющий решить вопрос о планарности графа за $O(n \log n)$ операций и затем упростили его и довели оценку до $O(n)$.

5. В теории сетей (транспортных, энергетических, связи и др.) большую роль играет задача о максимизации потока. Поток формально описывается ориентированным графом Γ с двумя отмеченными вершинами — источником I и стоком S . Каждое ребро r графа Γ характеризуется своей пропускной способностью $s(r)$. Поток f на Γ определяется потоками $f(r) \geq 0$ на каждом ребре r графа. Из содержательных соображений ясно, что суммарный поток, входящий в каждую вершину графа (кроме вершин I и S), равен суммарному потоку, вытекающему из этой вершины. Изучается величина потока — суммарный поток, выходящий из источника I (равный суммарному потоку, попадающему в сток S). Задача о максимальном потоке состоит в определении потока максимальной величины, ограниченного пропускными способностями ребер графа Γ .

Для решения задачи о максимальном потоке в 1962 г. был построен популярный в задачах сетевой оптимизации алгоритм Форда — Фолкерсона. Он позволяет на каждой итерации находить путь, вдоль которого поток наращивается. На практике алгоритм Форда — Фолкерсона используется успешно. Однако можно указать случаи, когда этот алгоритм не сходится. В связи с этим алгоритм подвергался многочисленным модификациям. Эдмондс и Карп построили алгоритм максимизации потока с оценкой числа операций $O(nm^2)$, где n — число вершин; m — число ребер графа Γ . Е. А. Диниц предложил алгоритм решения той же задачи за $O(n^2m)$ операций. А. В. Карзанов усовершенствовал алгоритм Диница и построил алгоритм с оценкой трудоемкости $O(n^3)$. Это при $m = O(n^2)$ всего на порядок выше нижней оценки числа начальных данных — $O(n+m) = O(n^2)$.

6. Помимо рассмотренных выше классов легко решаемых дискретных задач, в литературе рассматриваются и другие массовые проблемы, для решения которых разработаны полиномиальные алгоритмы. Однако число их невелико. Между тем дискретных задач, для которых до сих пор не найдено полиномиальных методов решения, очень много. Возникает вопрос: нельзя ли для этих задач построить относительно простые приближенные методы решения, которые, однако, гарантировали бы, что определяемая ими

погрешность решения не превысит заранее заданную величину. Оказывается, что для многих классов дискретных задач построение приближенного решения с заданной верхней границей погрешности столь же трудоемкая проблема, что и поиск точных методов. Доказано, например, что для задач линейного булевого программирования, задачи об оптимальной упаковке, задачи коммивояжера с произвольной матрицей расстояний и ряда других конструирование алгоритмов решения, обеспечивающих заданную верхнюю границу абсолютной погрешности ϵ , полиномиально эквивалентно получению точного решения. Временная сложность точного решения этих задач не более чем в полином раз от размерности превышает сложность построения приближенного решения с заданной верхней границей абсолютной погрешности.

Интересно отметить, что некоторое сужение класса матриц расстояний в задаче коммивояжера (до класса матриц, элементы которых удовлетворяют неравенству треугольника) уже позволяет строить полиномиальные приближенные алгоритмы для ее решения.

В последние годы выделены и другие классы дискретных задач, для которых не найдены эффективные методы точного решения, но могут быть построены полиномиальные приближенные методы, погрешность которых не превышает заданную величину. К таким задачам относятся, в частности, различные варианты составления расписания работы многопроцессорных систем — распределения n независимых заданий между m процессорами. Предполагается заданным время t_{ij} решения задачи i на процессоре j . Цель распределения — минимизировать суммарное время решения всех задач. Для приближенного решения подобных задач с малой относительной погрешностью разработаны алгоритмы с числом операций $O(n^2)$, $O(n \log n)$, $O(n)$ и потребной памятью $O(n)$.

Для задачи оптимального резервирования при ограничениях на габариты, вес и другие характеристики запасных элементов (для решения которой также нет эффективного точного алгоритма) построен достаточно эффективный приближенный алгоритм решения. Список аналогичных задач можно продолжить.

7. Трудности, возникающие при точном и приближенном решении задач дискретной оптимизации, заставляют менять подходы к постановкам таких задач и смягчать требования к решению. В предыдущем пункте предполагалось,

что приближенное решение должно обязательно удовлетворять всем ограничениям задачи, хотя и может не обеспечивать достижения оптимальной величины целевого функционала. Возможности дискретной оптимизации можно расширить, если допустить и малые невязки в условиях задачи. Эффективные приближенные алгоритмы, допускающие малые нарушения ограничений задачи, предложены для некоторых задач автоматизации проектирования радиоэлектронной аппаратуры.

Для задач булевого программирования с очень большим числом переменных (порядка 10^3 и больше), но весьма малым количеством ограничений (1—3) разработаны перспективные методы, эффективно срабатывающие «почти всегда» [4].

В [11] предложен другой (более грубый) подход к конструированию приближенных решений задач дискретной оптимизации, «оптимальных почти всегда». Это весьма простые методы, которые, однако, не гарантируют качества решения каждой индивидуальной задачи массовой проблемы. При некоторых нежестких требованиях к диапазону изменения параметров условий эти методы оказываются асимптотически оптимальными, т. е. при увеличении размерности n задачи доля точно решаемых задач увеличивается и стремится к единице при $n \rightarrow \infty$.

В [11] приведены простые алгоритмы для получения оптимального почти всегда решения задачи коммивояжера и некоторых других массовых дискретных задач. В [50] предложен асимптотически оптимальный алгоритм для решения обобщенной задачи о соединении городов. Этот же алгоритм может быть использован для асимптотически оптимального решения многоиндексных транспортных задач специального вида и для эффективной аппроксимации многомерного распределения вероятностей по ограниченному статистическому материалу.

Асимптотический подход представляет интерес при необходимости «в среднем хорошо» решить большое число индивидуальных задач массовой проблемы большой размерности. Этот подход, однако, не дает никаких гарантий при решении каждой конкретной задачи.

Подробные обзоры приближенных методов дискретной оптимизации и обсуждение их трудоемкости содержатся в монографии [39] и в статье [10].

Вычислительная сложность переборных задач

1. Как мы видели, известные точные методы решения комбинаторных задач, в частности задач дискретной оптимизации, да и многие приближенные методы оказываются малоэффективными при решении задач немалой размерности. При конечном числе альтернатив можно всегда выбрать лучший вариант, перебирая альтернативы и сравнивая их между собой. Однако при немалом числе допустимых вариантов перебор требует астрономического объема вычислений. Для алгоритмиста и вычислителя крайне важно выяснить, есть ли надежда построить достаточно общие и эффективные методы точного или приближенного решения массовых вычислительных задач дискретной математики. Если таких возможностей нет, то стратегия вычислителя меняется. В такой ситуации следует, по-видимому, разбивать массовые дискретные задачи на все более узкие классы и строить более эффективные методы, основанные на изучении особенностей частных подклассов задач.

В начале 70-х годов были опубликованы работы Л.А. Левина [27], Р. Карпа [17] и С. Кука [25], посвященные вычислительной сложности так называемых переборных задач. Переборные задачи — это, грубо говоря, классы задач (массовые задачи), каждая из которых имеет конечное число вариантов решения и может быть решена (во всяком случае, теоретически) с помощью полного перебора вариантов. По крайней мере один из параметров (например, размерность n) массовой переборной проблемы неограничен. Таким образом, число индивидуальных задач в переборной проблеме бесконечно. Индивидуальные задачи переборной проблемы различаются исходными данными — значениями числовых параметров условий. Такие известные задачи, как задача коммивояжера, различные задачи теории расписаний, задача о покрытии, задача о раскраске, трехиндексная задача о назначениях, и многие другие являются задачами переборного типа.

Проблема анализа переборных задач заключается не в том, чтобы найти конечный алгоритм решения любой задачи класса (таким алгоритмом является тривиальный перебор — алгоритм экспоненциальной вычислительной сложности), а в том, чтобы найти эффективный — полиномиальный — алгоритм, решающий каждую (без исключений) задачу класса достаточно быстро.

Переборные задачи возникают в различных областях

дискретной математики и формулируются на языке той области, в которой они используются. Для удобства анализа целесообразно кодировать дискретные задачи (математические объекты и отношения между ними) словами в некотором алфавите. Обычно для этого используется двоичная система счисления. Число символов, необходимых для записи условий задачи, определяет ее размерность.

Работа всякого алгоритма составляется из отдельных шагов. Каждый шаг требует определенного числа элементарных операций. Число шагов или число элементарных операций, необходимых для вычисления решения, определяет временную сложность работы алгоритма. Обычно заранее устанавливаются, в каких единицах (в числе шагов алгоритма или в количестве тех или иных операций) оценивается вычислительная сложность. За редким исключением, число элементарных операций на шаге существенно не зависит от размерности задачи и порядок временной сложности не зависит от того, в каких категориях она оценивается.

Различным индивидуальным задачам, составляющим данную переборную задачу, отвечает разная входная информация и разное число шагов алгоритма. В качестве характеристики входной информации x естественно брать длину $l(x)$ слова, которым она кодируется. Временная сложность алгоритма A на задаче x описывается функцией $f_A(x)$, определяющей число шагов (или элементарных операций), необходимых алгоритму A для решения задачи x класса.

2. Уже указывалось, что основной вопрос, возникающий при теоретическом исследовании переборной задачи, — это вопрос о том, является ли она полиномиально разрешимой.

Состояние дел здесь до появления современной теории вычислительной сложности было таково. Были известны отдельные переборные полиномиально разрешимые задачи (некоторые из легко решаемых задач описаны выше). Имелось также много практически важных задач, для решения которых, несмотря на все усилия, до сих пор не найдены полиномиальные алгоритмы (некоторые из этих задач будут описаны далее). При этом каждая из задач второй группы изучалась более или менее самостоятельно.

Что же нового внесла в этот круг вопросов теория вычислительной сложности? Прежде всего она выявила единую основу всех трудностей, возникающих в теории переборных задач. Было доказано существование универсальных переборных задач — таких задач U , из полиномиальной разре-

шимости которых следует полиномиальная разрешимость любой переборной задачи. В классе хорошо интерпретируемых в содержательных терминах задач универсальные задачи представляют собой «максимально сложные» задачи.

Показано, кроме того, что много естественных переборных задач (например, задача коммивояжера, задача о раскраске, задача о покрытии, ряд классических задач теории расписаний и др.) являются универсальными. В работе Л. А. Левина [27] было приведено шесть универсальных задач. В статье Р. М. Карпа [17] содержится 20 таких задач. В последние 10 лет число задач, универсальность которых установлена, быстро увеличивалось. Сейчас известно несколько тысяч универсальных задач.

Расширение списка универсальных задач представляет не только теоретический интерес, но и важно с практической точки зрения. Если бы удалось придумать полиномиальный алгоритм для любой универсальной задачи, скажем задачи коммивояжера, то оказалось бы возможным построить (причем эффективно) и алгоритмы решения всех без исключения переборных задач. Если же, наоборот, удалось бы доказать, что эта задача не является полиномиально разрешимой, то огромное число практически важных задач оказалось бы таким же. Найти полиномиальный алгоритм для какой-нибудь одной универсальной задачи не проще, чем для произвольной переборной задачи.

Подчеркнем, что центральный теоретический вопрос, возникающий в связи с изложенным, вопрос о том, существуют ли полиномиально неразрешимые переборные задачи, до сих пор не решен. Практическое значение решения этого вопроса трудно переоценить. Первая альтернатива означала бы принципиальную безнадежность попыток эффективно решать (по крайней мере решать точно) огромное семейство важных для практики задач. Вторая же альтернатива, напротив, сразу ликвидировала бы все теоретические трудности в решении переборных задач.

Высказывается гипотеза, что для универсальных задач вообще не существует эффективных методов решения. Неудачные попытки решения классических универсальных задач, предпринимаемые многими специалистами вот уже несколько десятилетий, заставляют многих ученых все больше склоняться к мысли о справедливости этой гипотезы. Правда, несколько неожиданный результат, полученный недавно в смежной области (о нем подробнее в следующем параграфе), насторожил многих специалистов,

и их высказывания по поводу возможности построения эффективных методов решения универсальных задач стали более осторожными.

3. Понятия переборной и универсальной задач были введены Л. А. Левиным. В работах Р. М. Карпа, С. Кука и других по вычислительной сложности использовались другие понятия и несколько иная терминология. В категориях и терминах работ Карпа и Кука естественней использовать важное для теории вычислительной сложности понятие сводимости задач, возникшее в математической логике. Техника сводимости позволяет относительно элементарными средствами расширить класс универсальных задач. Говорят, что переборная задача A сводится к переборной задаче B , если метод решения задачи B можно преобразовать в метод решения задачи A . Сводимость называется полиномиальной, если подобное преобразование можно произвести за полиномиальное число шагов (или элементарных операций). В работах Карпа и Кука рассматривается временная сложность решения переборных задач. Предложенная ими методология анализа переборных задач оказалась прогрессивной и позволила существенно расширить класс обнаруженных универсальных задач и исследовать их с других точек зрения.

Различают два типа переборных задач: задачи дискретной оптимизации (экстремальные задачи) и задачи распознавания свойств. В экстремальных задачах задана область G изменения дискретной переменной x и целевая функция $f(x)$. Требуется найти значение $x \in G$, при котором $f(x)$ достигает экстремума. В задачах распознавания свойств (или, как их иногда называют, задачах типа «да — нет») требуется определить, обладает ли функция $f(x)$ дискретного переменного x некоторым фиксированным свойством. Естественные переборные задачи, представляющие практический интерес, можно формулировать как в терминах задач оптимизации, так и в терминах задач распознавания свойств. Строго говоря, излагая концепцию Кука — Карпа, следовало бы говорить только о задачах распознавания свойств. Но мы сильно не погрешим против истины, если в последующих рассуждениях не всегда будем выделять задачи распознавания свойств из классов переборных задач.

Подход Кука — Карпа к решению переборных задач требует ввода недетерминированных машин или недетерминированных алгоритмов.

Возможные траектории процесса решения переборной

задачи можно представить в виде дерева, ветви которого отвечают последовательности шагов различных решающих алгоритмов. Массовая задача распознавания свойств считается решенной, если для каждой индивидуальной задачи будет найдена ветвь дерева, которая приводит к ответу «да», или если будет установлено, что все ветви приводят к ответу «нет». Последнее равносильно тому, что на дереве решения обратной задачи, «верно ли, что объект не обладает указанным свойством», будет найдена ветвь, которая приводит к ответу «да». Время, требуемое для решения индивидуальной задачи класса, определяется числом шагов кратчайшей траектории, приводящей к ответу «да».

Работу недетерминированного алгоритма (или недетерминированной машины) можно рассматривать как параллельную работу многих детерминированных алгоритмов (детерминированных машин), отвечающих каждой возможной траектории процесса решения. Каждая машина выполняет последовательность шагов соответствующей траектории процесса решения до тех пор, пока не будет получен ответ «да» или пока не окажется, что дальнейшие шаги невозможны. Другими словами, можно считать, что процесс решения задачи распознавания свойств на недетерминированной машине состоит из двух этапов: «угадывания» траектории процесса, приводящей кратчайшим путем к ответу «да», и проверки правильности догадки. По заданной индивидуальной задаче «угадывается» ветвь дерева возможных процессов решения — наиболее экономный из алгоритмов, распознающих заданное свойство, а затем рассматриваемая индивидуальная задача подается на детерминированную машину, алгоритм функционирования которой соответствует угаданной ветви. Временная сложность решения переборной задачи на недетерминированной машине определяется как минимальное число шагов, используемое «угаданной» ветвью дерева возможных процессов решения.

Рассмотрим в качестве примера задачу коммивояжера. Даны n пунктов и расстояния между ними. Нужно построить маршрут минимальной протяженности, позволяющий обойти все пункты, не заходя в один и тот же дважды. Это экстремальная задача, но ее можно свести к нескольким задачам распознавания свойств: существует ли проходящий через все пункты маршрут длины не больше заданной величины k . Для задачи коммивояжера неизвестен полиномиальный алгоритм ее решения. Допустим, однако, что для некоторой индивидуальной задачи каким-то образом полу-

чены основания предполагать, что существует искомый маршрут, длиной, не превышающей k . Чтобы убедиться в том, что это действительно так, достаточно предъявить этот маршрут, проверить, проходит ли он через все пункты ровно по одному разу, вычислить его длину и сравнить ее с заданной величиной k . Как легко установить, временная сложность процедуры проверки «угаданного» решения ограничена полиномом от n .

Полиномиальная проверяемость не влечет полиномиальной разрешимости. Утверждая, что за полиномиальное время можно проверить ответ «да», мы не учитываем время на «угадывание» нужной ветви дерева возможных решений.

Будем говорить, что недетерминированная машина решает массовую задачу распознавания Π за полиномиальное время, если существует полином такой, что для любой индивидуальной задачи $I \in \Pi$ найдется некоторая догадка, длительность проверки которой на детерминированной машине ограничена этим полиномом от размерности задачи.

4. Введем теперь два класса задач распознавания свойств (учитывая приведенную ранее оговорку, будем говорить о двух классах переборных задач). Класс P (Polynomial) обозначает класс переборных задач, решение которых может быть проведено на детерминированной машине за полиномиальное время. Класс NP (Nondeterministic Polynomial) обозначает класс переборных задач, полиномиально разрешимых на недетерминированных машинах.

Основная проблема в теории переборных задач заключается в выяснении вопроса: «Существуют ли задачи, которые содержатся в классе NP и не содержатся в классе P ?»

Эта проблема в теории до сих пор не решена. Кук [25] показал, что класс NP содержит некий подкласс задач, которые он назвал NP -полными, в некотором смысле «самые трудные» задачи. Если для какой-либо NP -полной задачи будет найден полиномиальный алгоритм решения на детерминированной машине, то окажется возможным построение полиномиального алгоритма для любой задачи класса NP и, следовательно, $P = NP$.

Таким образом, NP -полные задачи по Куку и Карпу представляют собой универсальные переборные задачи по Левину.

Проблема « $P = NP$?» является одной из важнейших проблем теории вычислительной сложности.

Полиномиальная разрешимость задач линейного программирования

1. Уже указывалось, что для подавляющего большинства переборных задач, для которых до сих пор не было найдено эффективных методов решения, доказана их NP -полнота (универсальность). Имеются, однако, и единичные обратные примеры. Для некоторых задач, которые предполагались универсальными, в последнее время построены полиномиальные методы решения. Один такой пример, вызвавший своего рода сенсацию (и породивший сомнения по поводу справедливости гипотезы $P=NP?$), стоит того, чтобы о нем рассказать.

Речь идет о задаче выяснения совместности системы линейных неравенств

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = 1, \dots, m.$$

Эта задача представляет собой частный случай традиционной задачи (нецелочисленного) линейного программирования и в принципе может быть решена любым конечным (т. е. обеспечивающим точное решение за конечное число арифметических действий) методом линейного программирования. Однако, как уже отмечалось, симплекс-метод, впрочем, как и все другие известные ранее конечные методы линейного программирования, гарантирует точное решение задачи лишь за число операций, экспоненциально растущее с ростом размерности задачи. Подчеркнем, что речь идет о гарантиях. Известные конечные методы на практике достаточно эффективны, но для каждого из них построены нетривиальные примеры задач с экспоненциальной сложностью вычислений.

Неоднократно ставился вопрос о том, является ли указанный факт дефектом методов или лежит в природе вещей. Последнее означает, что в принципе не существует конечных методов решения общей задачи линейного программирования, скажем, с полиномиальной по $n+m$ оценкой необходимого числа арифметических операций. Заметим, что речь шла об арифметических действиях, а не о трудоемкости в смысле современной теории вычислительной сложности. Дело в том, что линейные задачи с вещественными коэффициентами не могут быть закодированы конечными двоичными словами.

2. К всеобщему удивлению, относительно недавно

Л. Г. Хачияну [40] удалось достичь замечательного продвижения в этом направлении и, по существу, решить поставленный вопрос. Оговорка «по существу» связана с тем, что сам вопрос был слегка видоизменен, впрочем, без особого ущерба для его смысла. Вместо рассмотрения линейных задач с вещественными коэффициентами — бесконечными двоичными дробями Хачиян рассматривает задачи с рациональными коэффициентами — конечными двоичными дробями, или, что то же самое, задачи с целыми коэффициентами. Это сужение семейства задач абсолютно несущественно для приложений. Зато получаемая массовая задача попадает в сферу действия современной теории вычислительной сложности. В частности, можно спросить, является ли она полиномиально разрешимой. Этот вопрос теперь имеет несколько иной смысл, чем раньше, в своей классической постановке. С одной стороны, размерность задачи, таким образом, увеличилась, что увеличило шансы на существование эффективных методов решения (число шагов полиномиального метода теперь должно оцениваться сверху полиномом от большей величины, чем в классической постановке, поскольку размерность, которую здесь удобно характеризовать длиной ее кода, теперь уже не $n+m$, а сумма длин двоичных записей коэффициентов задачи). Эти шансы зато сократились за счет сужения возможностей методов (раньше им разрешалось выполнять неконструктивные операции — действия над бесконечными двоичными дробями, а теперь метод обязан быть алгоритмом в смысле математической логики, т. е. допускаются лишь конструктивные операции).

В новой постановке Л. Г. Хачиян решил поставленный вопрос, и, что самое удивительное, положительно. Он построил полиномиальный алгоритм установления совместности системы линейных неравенств с целыми коэффициентами и, кроме того, полиномиальный алгоритм решения общей задачи линейного программирования с целыми коэффициентами. Подчеркнем, что решение такой задачи отнюдь не обязано быть целочисленным вектором. Поэтому на первый взгляд кажется, что такая задача не укладывается в круг понятий дискретной математики. Это, однако, не так: среди решений рассматриваемой задачи (если они вообще имеются) обязательно есть и решение с рациональными координатами, и его можно закодировать, кодируя двоичными словами наборы из числителей и знаменателей соответствующих дробей.

Заметим, что в рассуждениях и выводах Л. Г. Хачияна существенно используется модифицированный метод центров тяжести (метод эллипсоидов) приближенного решения выпуклых экстремальных задач. Разработка этого метода вызвана проблемами информационной сложности, и о нем пойдет речь ниже в следующей главе. Там же будут сопоставлены основные понятия вычислительной и информационной сложности.

3. Опишем кратко ход рассуждений, лежащих в основе построения полиномиального алгоритма для выяснения разрешимости (в вещественных числах) системы линейных неравенств с целыми коэффициентами.

Рассмотрим систему $\sum_{j=1}^n a_{ij}x_j \leq b_i, i \in I$. Под размерностью α этой системы будем понимать сумму длин двоичных записей всех ее коэффициентов. Пусть $\varphi(x) = \max_{i \in I} \{ \sum_{j=1}^n a_{ij}x_j - b_i \}$. Идея построения полиномиального алгоритма, позволяющего выяснить разрешимость рассматриваемой системы неравенств, заключается в следующих замечаниях:

(1) Если система линейных неравенств совместна, то у нее есть решение в шаре V радиуса 2^α с центром в 0 и, следовательно, в этом случае $\min_V \varphi(x) \leq 0$.

(2) Если система несовместна, то $\min_V \varphi(x) \geq 2^{-\alpha}$.

Это значит, что для решения вопроса о совместности системы линейных неравенств достаточно оценить величину $s = \min_V \varphi(x)$ с погрешностью, например, не превышающей $\frac{1}{3} \cdot 2^{-\alpha}$.

Функция $\varphi(x)$ как максимум линейных форм выпукла, и ее минимизация на шаре V есть задача выпуклого программирования. Как видно из построений, приведенных в следующей главе, минимизация $\varphi(x)$ на V и оценка трудоемкости получения требуемого приближения к оптимуму могут быть проведены по методу эллипсоидов. Число итераций, необходимых для решения задачи по методу эллипсоидов (и число элементарных операций модификации метода для решения системы линейных неравенств), полиномиальным образом зависит от числа переменных n , стало быть, от α . Отсюда следует возможность построения полиномиального алгоритма для выяснения совместности линейных неравенств.

Развитие описанных идей привело к созданию «полиномиальных» алгоритмов для решения задач линейного и выпуклого квадратичного программирования с целыми коэффициентами.

4. Необходимо подчеркнуть, что описанные в настоящем параграфе результаты относятся к линейным задачам, у которых целочисленны лишь коэффициенты целевой функции и элементы матрицы условий. На компоненты искомого решения никаких условий целочисленности не накладывается. Не следует путать целочисленные в этом смысле задачи с задачами целочисленного линейного программирования (где оптимизация ведется на целочисленной решетке). Задача последнего типа универсальна, а для задач, рассмотренных в [40], универсальность не доказана. Поэтому результаты Л. Г. Хачияна не дают пока никакой существенной информации для решения центральной проблемы теории вычислительной сложности — проблемы « $P = NP?$ », т. е. вопроса о полиномиальной разрешимости универсальных задач перебора.

Труднорешаемые дискретные задачи

1. Список универсальных (NP -полных) дискретных массовых задач в настоящее время насчитывает тысячи задач, и с каждым днем этот список расширяется. Описания задач, для которых доказана NP -полнота, разбросаны по различным источникам. Наиболее полный перечень NP полных задач — более 300 задач, почти каждая из которых имеет несколько версий, — приведен в [12]. Мы здесь опишем несколько универсальных задач, к которым сводятся многие технические и экономические проблемы. Напомним еще раз, что речь идет о наиболее трудных переборных задачах. Если для какой-либо одной из них будет найден полиномиальный алгоритм решения (метод решения, вычислительная сложность которого растет как полином от размерности задачи), то этот алгоритм можно будет преобразовать и приспособить, сохраняя его полиномиальность по размерности, для любой другой переборной задачи и одна из важнейших проблем вычислительной математики « $P = NP?$ » будет решена.

2. Неоднократно упоминавшаяся выше задача коммивояжера — наиболее популярная универсальная задача —

имеет много различных экономических и технических интерпретаций. Пусть задано n пунктов и целочисленная матрица расстояний между ними $D = \|d_{ij}\|$ (d_{ij} — расстояние между пунктами i и j). Требуется найти маршрут минимальной длины обхода всех пунктов, не заходящий дважды в один и тот же пункт. Такова формулировка задачи коммивояжера в терминах экстремальных задач. В постановке, используемой в задачах распознавания свойств, предполагается, что задано еще целое число $k > 0$ и требуется установить, имеется ли маршрут длины, не превышающей k , позволяющий обойти все пункты, не посещая дважды какой-либо из них.

Задача коммивояжера остается универсальной даже в том случае, если все расстояния d_{ij} принимают одно из двух значений, например, 1 или 2.

К задаче коммивояжера полиномиально сводится задача о гамильтоновом цикле, играющая важную роль в теории графов и ее приложениях. Она формулируется следующим образом. Пусть Γ — граф с множеством вершин B и множеством ребер P . Простым циклом в графе Γ называется такая последовательность $\{b_1, b_2, \dots, b_k\}$ различных вершин из B , что отрезки, соединяющие пары вершин $\{b_i, b_{i+1}\}$, $1 \leq i < k$, и отрезок $\{b_k, b_1\}$ принадлежат P , т. е. являются ребрами графа Γ . Гамильтоновым циклом в Γ называется простой цикл, содержащий все вершины графа Γ . Задача о гамильтоновом цикле требует проверить, верно ли, что граф Γ содержит гамильтонов цикл.

Задача о гамильтоновом цикле полиномиально эквивалентна задаче коммивояжера и, следовательно, также является универсальной (NP -полной) задачей. Чтобы убедиться в этом, необходимо указать функцию ϕ , отображающую каждую индивидуальную задачу из массовой проблемы «гамильтонов цикл» (ГЦ) в соответствующую индивидуальную задачу массовой проблемы «коммивояжер» (КВ). Пусть граф Γ , определяемый множеством B из n вершин и множеством P ребер, означает фиксированную индивидуальную задачу из ГЦ. Соответствующая задача из КВ строится следующим образом. Множество пунктов Π совпадает с множеством вершин B графа Γ . Определим расстояние $d(b_i, b_j)$ между двумя пунктами $b_i, b_j \in \Pi$ так, что $d(b_i, b_j) = 1$, если отрезок $[b_i, b_j] \in P$, т. е. является ребром графа Γ , и $d(b_i, b_j) = 2$ — в противном случае. Примем число $k > 0$, определяющее задачу КВ в терминах распознавания свойств, равным n . Функция ϕ , устанавливающая своди-

мость индивидуальных задач ГЦ к индивидуальным задачам КВ, может быть вычислена за полиномиальное время.

Для вычисления $\frac{1}{2} n(n-1)$ расстояний $d(b_i, b_j)$ необходимо лишь выяснить, являются ли отрезки $[b_i, b_j]$ ребрами графа Γ . Строго говоря, для проверки полиномиальной сводимости ГЦ к КВ следовало бы еще показать, что Γ содержит гамильтонов цикл в том и только в том случае, если в соответствующей задаче КВ-имеется соединяющий все пункты из Π маршрут длиной не больше k . Но этот этап доказательства полиномиальной сводимости ГЦ к КВ мы опустим. Мы опустим также доказательство полиномиальной сводимости КВ к ГЦ.

3. В сетевых задачах транспортного типа часто приходится оценивать качество маршрута по нескольким критериям, например, по его длине и связанным с ним затратам. В связи с этим возникает задача выбора кратчайшего пути при заданных ограничениях на затраты.

В терминах распознавания свойств задача формулируется следующим образом. Задан граф Γ с множеством вершин V и множеством ребер P . В графе выделены две вершины: источник (И) и сток (С). Каждому ребру $r \in P$ графа Γ приводятся в соответствие два целых положительных числа — длина ребра $l(r)$ и затраты $s(r)$, связанные с перемещением по этому ребру. Заданы, кроме того, два целых положительных числа L и S . Требуется выяснить, существует ли в графе Γ путь из вершины И в вершину С, длина которого не превышает L при затратах не больше S . Это NP -полная задача.

NP -полнота задачи о кратчайшем пути сохраняется и в том случае, когда Γ — ориентированный граф.

Обе задачи, однако, разрешимы за полиномиальное время, если длины всех ребер графа Γ равны между собой или затраты, связанные с перемещением по разным ребрам, одинаковы. В этих случаях получаемая задача эквивалентна задаче о максимальном потоке, рассмотренной при изучении легкорешаемых задач.

К рассматриваемым задачам полиномиально сводится задача о разбиении множества A элементов a разной стоимости $s(a)$ ($s(a)$ — целые положительные числа при всех a) на два множества A_1 и A_2 ($A_1 \cap A_2 = \emptyset$, $A_1 \cup A_2 = A$) одинаковой стоимости. В задаче о разбиении следует, таким образом, установить, существует ли множество $A_1 \subseteq A$

такое, что

$$\sum_{a \in A_1} s(a) = \sum_{a \in A \setminus A_1} s(a).$$

Задача о разбиении — универсальная задача.

4. Еще один пример универсальной задачи — задача о минимизации времени передачи сообщения — состоит в следующем. Имеется множество пунктов и сеть связи, позволяющая пунктам — соседним узлам сети — непосредственно без ретрансляции (за один сеанс связи) передавать один другому сообщения.

Представим сеть в виде графа Γ с множеством V вершин, отвечающих пунктам, и множеством P ребер, отвечающих линиям связи. Пусть некоторое подмножество $V_0 \subset V$ пунктов связи располагает определенной информацией (сообщением), которую нужно передать всем пунктам сети. За один сеанс связи можно передать сообщение только в смежную вершину, т. е. в вершину, соединенную с источником информации ребром. Предполагается, что в течение сеанса связи можно передать сообщение из каждого пункта, обладающего информацией, только в один пункт, и, наоборот, каждый пункт, не располагающий информацией, может получить ее только из одного пункта. На очередном сеансе связи выбираются некоторые из пунктов (вершин графа), которые еще не получили сообщение. Для того чтобы они могли на этом сеансе получить информацию, они должны быть непосредственно связаны линией связи (ребром графа Γ) хотя бы с одним из пунктов, которые к этому моменту уже получили сообщение. В задаче требуется организовать передачу сообщения таким образом, чтобы оповестить все пункты за минимальное число сеансов связи. Это экстремальная постановка задачи. В терминах распознавания свойств задается еще число k сеансов связи и требуется выяснить, можно ли за число шагов (сеансов связи), не превосходящее k , передать, соблюдая приведенные выше условия, сообщение из исходного множества V_0 пунктов во все пункты сети.

Формально постановка этой задачи записывается следующим образом.

Требуется установить, существует ли такая последовательность множеств вершин и ребер графа Γ : $V_0, P_1, V_1, P_2, \dots, P_t, V_t$, что $V_i \subseteq V$, $P_i \subseteq P$, $V_t = V$ и для всех i , $1 \leq i \leq t$, выполнены условия:

1° ровно один конец каждого ребра из множества P_i принадлежит множеству вершин B_{i-1} ;

2° никакие два ребра из P_i не имеют общего конца;

3° $B_i = B_{i-1} \cup \{b : [a, b] \in P_i\}$, $a, b \in B$.

Доказано, что при любом заданном числе сеансов $t \geq 4$ задача о минимизации времени передачи сообщения NP -полная, а при $t=1$ для ее решения существует полиномиальный алгоритм.

Если исходное множество B_0 вершин содержит единственную вершину, задача остается NP -полной, но если при этом граф Γ — дерево (т. е. односвязен и не имеет циклов), то задача оказывается полиномиально разрешимой.

К рассмотренной модели полиномиально сводится задача о трехмерном сочетании, обобщающая задачу о двухмерном сочетании, известную в литературе под названием задачи о бракосочетании. Последняя формулируется следующим образом: имеется n холостых мужчин и столько же незамужних женщин. Каждая женщина и каждый мужчина составляют список приемлемых партнеров. По этим спискам формируется список всех пар мужчин и женщин, согласных вступить в брак друг с другом. В задаче требуется установить, можно ли совершить n бракосочетаний так, чтобы каждый получил приемлемую супругу (или супруга) и чтобы при этом никто не вступал в брак дважды.

Оказывается, задачу о двухмерном сочетании можно отнести к легкорешаемым. Для ее решения существует полиномиальный алгоритм. А вот если задачу обобщить и перейти от двухмерного сочетания к трехмерному, то задача станет труднорешаемой — NP -полной. В приведенных терминах задача о трехмерном сочетании формулируется так. Имеется по n представителей трёх разных «полов» (например, как у А. Азимова в «Сами боги»: левник, правник и централь) и списки, отвечающие «трехмерному сочетанию», приемлемому для каждого участника любой потенциальной триады. Требуется выяснить, можно ли осуществить n трехмерных сочетаний, приемлемых для каждого участника любого «пола» и чтобы при этом ни один представитель не участвовал в двух триадах одновременно.

Ясно, что многие трехиндексные задачи транспортного или распределительного типа укладываются в схему модели о трехмерных сочетаниях.

5. При организации машинных вычислений возникает задача динамического распределения памяти. Сущность этой задачи в следующем. Пусть имеется множество A

данных. Предполагаются заданными длина элемента $a \in A$ — целое положительное число (число букв слова a), моменты поступления слова a и окончания работы со словом a — целые положительные числа. Задача заключается в том, чтобы найти минимальный объем памяти, обеспечивающий распределение данных в памяти, учитывающее динамику их поступления и пребывания в системе.

Это экстремальная постановка задачи. В терминах распознавания свойств задача формулируется следующим образом. Задано целое положительное число k — объем памяти вычислительного устройства. Требуется выяснить, можно ли поставить в соответствие каждому слову $a \in A$ номер ячейки $\sigma(a)$ в устройстве памяти $\sigma(a) \in \{1, 2, \dots, k\}$, с которой следует начинать запись слова, таким образом, чтобы выполнялись следующие условия:

1° запись любого слова $a \in A$, начатая в ячейке $\sigma(a)$, заканчивается в ячейке с номером, не превышающим k ;

2° если два интервала в устройстве памяти, в которых записаны слова a_1 и a_2 , пересекаются, то момент окончания работы с одним из этих слов наступает не позже момента поступления другого слова.

Задача динамического распределения памяти *NP*-полная. Она остаётся *NP*-полной даже в том случае, если длина всех слов — элементов данных, подлежащих записи в памяти, принимает одно из двух значений 1 или 2.

К задаче о динамическом распределении памяти полиномиально сводится версия задачи о разбиении множества, рассмотренной ранее и называемая «3-разбиение», или задача разбиения на тройки. Задано множество A из $3n$ слов и целое положительное число k . Длина слова a равна

$l(a)$ и при этом $\frac{k}{4} < l(a) < \frac{k}{2}$, а общая длина записи

всех слов множества A равна nk . Требуется выяснить, можно ли разбить множество A на n непересекающихся подмножеств A_1, A_2, \dots, A_n так, чтобы суммарная длина записи элементов каждого подмножества равнялась k .

Из ограничений на длины $\frac{k}{4} < l(a) < \frac{k}{2}$ и требования

$\sum_{a \in A_i} l(a) = k$ следует, что $\frac{|A_i|}{4} < 1 < \frac{|A_i|}{2}$, где $|A_i|$ —

число элементов в множестве A_i . Отсюда $|A_i| = 3$, т. е. каждое из множеств A_i должно содержать ровно три элемента множества A . Задача разбиения слов на тройки одинаковой длины *NP*-полная задача.

6. Оказывается, и традиционная задача составления учебного расписания представляет собой NP -полную задачу. Сформулируем эту задачу. Пусть задано множество T — рабочих часов, множество P — преподавателей и множество D — учебных дисциплин. Для каждого преподавателя $p \in P$ указано подмножество рабочих часов $A(p) \subseteq T$, приемлемых для него. Для каждой дисциплины $d \in D$ задано подмножество рабочих часов $B(d) \subseteq T$, допустимых для этой дисциплины (обусловленное, например, внеучебными факторами), и наконец, для каждой пары (p, d) — «преподаватель — дисциплина» — предписана требуемая нагрузка $r(p, d)$ — число часов, которое преподаватель p должен отвести дисциплине d . Требуется установить, существует ли расписание, учитывающее учебную программу, пожелания преподавателей, требуемую нагрузку и заданные внеучебные факторы. Запишем задачу в формальных терминах.

Обозначим через f булеву функцию трех переменных p, d, t такую, что $f(p, d, t) = 1$ означает, что преподаватель p проводит занятия по дисциплине d в час t .

Задача заключается в том, чтобы выяснить, существует ли булева функция $f(p, d, t)$, удовлетворяющая следующим условиям:

1° $f(p, d, t) = 1$ в том и только в том случае, если $t \in A(p) \cap B(d)$, т. е. лекция по дисциплине d назначается преподавателю p в час t тогда и только тогда, когда это приемлемо для преподавателя и допустимо для дисциплины;

2° для каждого часа $t \in T$ и преподавателя $p \in P$ существует не более одной дисциплины $d \in D$, такой, что при этом $f(p, d, t) = 1$;

3° для каждого часа $t \in T$ и дисциплины $d \in D$ можно указать не более одного преподавателя $p \in P$, так, чтобы $f(p, d, t) = 1$;

4° для каждой пары «преподаватель — дисциплина» $(p, d) \in P \times D$ существует ровно $r(p, d)$ значений t (рабочих часов), для которых $f(p, d, t) = 1$.

Задача составления учебного расписания остается NP -полной даже при следующих упрощениях, предполагающих, что всего рабочих часов $|T| = 3$, все эти часы допустимы для всех дисциплин, т. е. $B(d) = T$ для всех $d \in D$, и требуемые нагрузки могут равняться одному из двух чисел 0 или 1.

Общая задача составления учебного расписания становится полиномиально разрешимой в одном из двух случаев:

(а) если число часов, приемлемых для каждого преподавателя, не превышает двух ($|A(p)| \leq 2$);

(б) если любой час $t \in T$ приемлем для любого преподавателя и допустим для каждой дисциплины.

7. В заключение настоящего параграфа сформулируем без обсуждения еще несколько универсальных переборных задач.

(1) **Общая задача целочисленного линейного программирования.** Даны целочисленная $m \times n$ матрица $A = \|a_{ij}\|$, целочисленные m -мерный вектор $b = (b_1, \dots, b_m)$ и n -мерный вектор $c = (c_1, \dots, c_n)$. Требуется среди всех решений системы неравенств

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, \dots, m$$
 найти целочисленный n -мерный вектор $x = (x_1, \dots, x_n)$, на котором линейная форма $\sum_{j=1}^n c_j x_j$ достигает максимума.

Более узкая задача булева линейного программирования (задача целочисленного линейного программирования, в которой все x_i могут принимать лишь значения 0 или 1) также является универсальной переборной задачей. Однако для решения «почти всех» задач булева программирования существует полиномиальный метод. Оказывается, «почти все» задачи $\{0, 1\}$ -программирования можно решить на детерминированной машине Тьюринга за квадратичное время от длины записи исходной информации.

(2) **Решение системы булевых неравенств** может рассматриваться как подкласс задач целочисленного линейного программирования. Дана система неравенств $\sum_{j=1}^n a_{ij} x_j \leq b_i$,

$i = 1, \dots, m$, где элементы a_{ij} матрицы A состоят из нулей и положительных и отрицательных единиц, а компоненты b_i вектора b — нули и единицы. Требуется установить, имеет ли эта система булево решение, т. е. существует ли вектор $x = (x_1, \dots, x_n)$ из нулей и единиц, удовлетворяющий этой системе неравенств.

(3) К частной задаче целочисленного линейного программирования сводится и задача о разрешимости диофантова

уравнения. Дано уравнение $\sum_{j=1}^n a_j x_j = b$, где a_j, b — целые числа. Требуется выяснить, имеет ли оно булево решение — вектор x с компонентами 0 и 1. Заданное уравнение можно

представить в виде системы двух неравенств $\sum_{j=1}^n a_j x_j \leq b$,
 $-\sum_{j=1}^n a_j x_j \leq -b$.

(4) **Задача о клике.** Дан граф Γ и целое число $k > 0$. Требуется установить, существуют ли полный подграф (т. е. подграф, любые две вершины которого соединены ребром), имеющий k вершин. Если граф Γ характеризует организационную структуру, то задача требует определить, можно ли выделить сеть (клик) из k вершин — организационных единиц, непосредственно связанных друг с другом.

(5) **Задача об упаковке множеств.** Дано семейство множеств $\{A_i\}$ и целое число $k > 0$. Требуется установить, имеется ли в $\{A_i\}$ подсемейство, состоящее из k попарно не пересекающихся множеств.

(6) **Задача о покрытии множествами.** Дано семейство множеств $\{A_i\}$ и целое число $k > 0$. Требуется установить, существует ли в $\{A_i\}$ подсемейство $\{B_i\}$, содержащее не более k множеств, объединение которых совпадает с объединением всех множеств исходного семейства.

(7) **Задача о раскраске.** Дан граф Γ и целое число $k > 0$. Требуется установить, можно ли раскрасить вершины графа в k цветов так, что никакие две смежные вершины не будут окрашены в один цвет.

(8) **Задача Белмана — Джонсона составления расписания.** Даны числа t_i и τ_i , $i=1, \dots, n$, — продолжительности обработки i -го изделия соответственно на 1-й и 2-й машинах. Каждое изделие проходит обработку последовательно на каждой машине. Имеются емкости для хранения частично обработанных деталей. Заданы директивные сроки T_i и число $k > 0$. Требуется установить, имеется ли расписание, гарантирующее выполнение директивных сроков при общем времени обработки деталей, не превышающем k .

(9) **Задача Форда — Фалкерсона об организации параллельных работ.** Даны числа τ_i — продолжительности непрерываемых работ и t_{ij} — времени перехода от работы i к работе j , число s — одинаковых исполнителей, выполняющих работы параллельно, и число $k > 0$. Требуется установить, имеется ли распределение работ между исполнителями, при котором общее время выполнения работ не превышает k .

(10) **Задача о покомпонентных векторных неравенствах.** Даны два множества n -мерных векторов с целочисленными

компонентами $X = \{x_1, \dots, x_s\}$ и $Y = \{y_1, \dots, y_r\}$. Требуется установить, существует ли такой n -мерный вектор z с целочисленными компонентами, что число векторов $x_i \in X$, все компоненты которых не превосходят соответствующих составляющих z , не меньше числа векторов $y_j \in Y$, все компоненты которых не превосходят соответствующих составляющих z .

Задача остается *NP*-полной даже в том случае, если все составляющие векторов x_i и y_j равны 0 или 1.

В терминах перечисленных универсальных задач формулируются многие проблемы планирования, управления и проектирования.

Заключение

1. С некоторых пор не только физики и представители других естественных наук, но и математики начинают склоняться к целесообразности описания реальных явлений не на языке математики бесконечного и непрерывного, а на языке математики конечного и дискретного. В идеализированном описании реального мира все чаще предполагается некоторый уровень зернистости. Математики, уделяющие внимание приложениям, начинают смотреть на разностные уравнения, описывающие физические процессы, не как на приближения к точным моделям, а как на модели, отражающие сущность дискретной вселенной, клеточного автоматного пространства, в котором каждая клетка имеет только конечное число состояний. Появляются соображения о целесообразности отказаться от промежуточных непрерывных моделей и изучать реальные явления в их дискретном описании. Человеческий мозг оперирует в существенном по дискретному принципу, хотя математику, по-видимому, в силу исторического развития математики часто привычнее рассуждать в терминах гладких конструкций.

Относительно несложные (в любом используемом здесь смысле) явления и процессы, допускающие наглядное представление, легче воспринимать в непрерывных терминах, чем в комбинаторных. «Сглаживание» дискретных рядов — переход от дискретного множества точек к усредняющей их непрерывной (гладкой) поверхности — обычно используемый на практике прием приближенного выявления закономерностей явлений и процессов. Качественный анализ свойств относительно простых объектов и предсказание их

состояния и поведения удобней проводить в непрерывных понятиях и терминах. В сравнительно несложных задачах непрерывная математика позволяет получить аналитическим путем явные решения и обеспечивает, таким образом, установление наглядной зависимости вариации решения от изменения параметров условий задачи. Однако это достоинство непрерывной математики утрачивается по мере усложнения задач, анализ которых невозможен без численного решения каждой индивидуальной задачи класса.

Первоначальные закономерности в механике, физике, астрономии были связаны с линейными зависимостями. Уточнение закономерностей потребовало учета нелинейностей. Аппроксимация нелинейных эффектов обычно производится квадратичными или полиномиальными зависимостями более высоких степеней или аналитическими функциями. Во всех этих случаях часто удается получать явное решение задач, связанных с учетом соответствующих закономерностей. Количественными манипуляциями с этими классами функций, собственно, исчерпываются конструктивные достижения непрерывной математики, не требующие трудоемких машинных вычислений. Имеются, правда, еще выпуклые непрерывные задачи (и некоторые их расширения), в которых решение хоть и не может быть представлено в явном виде, но которые, однако, не требуют чрезмерно трудоемких вычислений. В приложениях (особенно в проблемах принятия решений) вряд ли могут быть указаны другие достаточно общие и широкие классы задач непрерывной математики, количественный анализ которых не требовал бы громоздких вычислений.

Возможности аналоговой вычислительной техники неизмеримо более скромны, чем возможности универсальных цифровых вычислительных машин. В терминах непрерывной математики ставятся и исследуются относительно «хорошие» задачи. В терминах дискретной математики требования к формализации задачи менее жесткие. Поэтому во всех проблемах, связанных с численным анализом, особенно в проблемах выбора решений, намечается тяготение к комбинаторным методам исследований и стремление к описанию реальных явлений дискретными моделями, минуя промежуточный этап их непрерывного описания. По-видимому, этим и объясняется внимание, которое в последние десятилетия все больше уделяется дискретным моделям прикладной математики.

Классификация задач по различным мерам сложности

также оказывается более естественной для дискретных описаний, чем для непрерывных. Дискретные задачи принятия решений имеют четкую алгебраическую структуру. Способы кодирования задачи для организации вычислений очевидны и естественны. В непрерывных задачах проблема кодирования не простая — далеко не каждый подход к ней сохраняет непрерывные классические свойства задачи. Отсюда и более фундаментальные результаты теории алгоритмической и вычислительной сложности, конструируемые в понятиях дискретной математики.

Математики всегда стремились создавать эффективные методы решения возможно более широких классов задач. Теория сложности показала, что общность и эффективность — противоречивые требования к методу решения. Для весьма широких классов задач может вообще не существовать единого алгоритма решения. А в случае алгоритмической разрешимости класса задач трудоемкость решения нередко убывает, если разбить класс задач на подклассы и, пользуясь их спецификой, разрабатывать для каждого из них эффективные методы анализа. Как мы видели, многие из этих проблем решены для задач дискретной математики. Еще больше задач ждет своего решения.

2. Вычислительная сложность, как она была определена в этой главе, представляет собой минимаксную оценку класса задач. Она оценивается минимальным объемом ресурсов, необходимых для решения **любой** задачи класса. С практической точки зрения часто больший интерес представляет средняя оценка ресурсов, необходимых для решения задач класса. Однако ее вычисление предполагает наличие некоторой информации об априорном распределении задач класса. Такой информацией далеко не всегда располагает исследователь. Тем не менее может оказаться, что для широкого класса часто встречающихся на практике распределений оценка сложности не зависит или слабо зависит от характеристик конкретного распределения. В таких ситуациях удастся в отдельных случаях оценить среднюю трудоемкость метода и получить больше оснований для прогнозирования затрат ресурсов, необходимых для решения массовой задачи. Так обстояло, например, дело с оценкой симплексного метода решения задач линейного программирования. Практика свидетельствует о высокой эффективности метода. Между тем еще четверть века назад были построены примеры задач линейного программирования, решение которых с помощью симплекс-метода требовало

почти полного перебора вершин многогранника условий. Следовательно, по минимаксной оценке симплекс-метод экспоненциально сложен. Однако совсем недавно (в 1982—1983 гг.) появились работы, в которых обоснована полиномиальная (с невысокой степенью полинома от размерности и числа ограничений) средняя трудоемкость симплекс-метода при естественных нежестких предположениях относительно априорного распределения задач в классе. Таким образом, устранено расхождение между теоретическими и экспериментальными результатами.

Следующим шагом в этом направлении является статистическая оценка сложности, учитывающая частоту появления «плохих» задач класса. Следует считать перспективными дальнейшие исследования по оценке средней и статистической сложности классов задач и трудоемкости методов. Это относится не только к дискретным задачам, но и к рассматриваемым в следующей главе непрерывным задачам оптимизации.

ИНФОРМАЦИОННАЯ СЛОЖНОСТЬ

Сложность оптимизации

В настоящей главе мы изучаем сложность решения задач оптимизации. Оптимизация в том или ином смысле является основой принятия рациональных решений. Каждое решение, особенно если его последствия влияют на наше будущее, должно быть наилучшим в некотором смысле, соответствующим специфике ситуации. В практике принятия решений содержательный смысл задачи часто определяет характеристику решения (или по крайней мере структурные свойства характеристики), оптимизация которой соответствует наилучшему выбору. Нередки также случаи, когда установление критерия эффективности предоставляется лицу, принимающему решение, имеющему опыт в постановке и анализе задач в соответствующей области и несущему ответственность за последствия принятых решений. Однако в достаточно ответственных ситуациях определение понятия «лучшее решение» представляет собой чрезвычайно сложную методологическую проблему, исследование которой далеко не всегда является предметом чисто математических изысканий. Здесь может оказаться необходимым учет многих критериев качества, определяемых различными требованиями к решению. Зачастую реше-

ние затрагивает интересы многих лиц или коллективов, что не может не сказаться на понятии «лучшее решение». Наконец, в случаях, когда общее решение проблемы определяется частными решениями отдельных участников процесса выбора решений, возникают конфликтные ситуации, разрешение которых требует переговоров и установления компромисса. Во всех этих проблемах пока еще больше неясного, чем готовых рецептов. Обсуждение связанных с ними вопросов остается за рамками настоящей работы. Мы здесь исходим из того, что после согласования понятий «равновесие», «компромисс» или «справедливость» удастся, как правило, сформировать некоторый критерий (или указать структурные свойства критерия), к оптимизации которого сводятся конструктивные подходы к выбору «равновесных», «компромиссных» или «справедливых» решений.

Так или иначе, оптимизация (условная или безусловная) обычно является основой или составной частью проблемы принятия решений. Поэтому оценка сложности задач оптимизации в зависимости от размерности задачи и допустимой погрешности представляет несомненный практический интерес.

2. Цель настоящей главы — оценить трудоемкость решения массовых задач оптимизации (различных классов задач оптимизации) и получить, таким образом, представление о сложности принятия решений в ситуациях, когда свойства критерия качества решения и области допустимых решений уже установлены. Собственно говоря, предыдущая глава, посвященная вычислительной сложности, охватывала в некотором смысле и эту проблематику. Однако изложенная там методология позволяет оценивать лишь трудоемкость решения дискретных задач оптимизации, условия которых фиксированы в виде набора параметров. Здесь мы будем рассматривать непрерывные экстремальные задачи — достаточно общие модели математического программирования, не укладывающиеся в схему дискретной оптимизации. Классы непрерывных экстремальных задач, как правило, не характеризуются набором параметров. Исключение составляют задачи типа линейного или квадратичного программирования, для которых теория информационной сложности дает весьма грубые оценки. Трудоемкость решения таких задач естественней оценивать в рамках теории вычислительной сложности. Излагаемая в настоящей главе теория информационной сложности приспособлена к оценке сложности классов задач оптимизации.

ции, описываемых такими аморфными характеристиками, как степень гладкости, выпуклость, мера обусловленности и др. Методология вычислительной сложности не оперирует подобными понятиями и не может быть использована для оценки трудоемкости решения соответствующих классов задач.

В главе существенно используются материалы А. С. Немировского и Д. Б. Юдина по постановкам задач и результатам теории информационной сложности.

Постановка задачи и основные определения

1. Теория информационной сложности изучает потенциальные возможности численных методов оптимизации для различных массовых условных экстремальных задач — для классов задач (недискретного) математического программирования.

Напомним, что в задаче математического программирования требуется вычислить n -мерный вектор x , оптимизирующий (обращающий в минимум или в максимум в зависимости от содержательной постановки задачи) критерий качества решения $f_0(x)$ при соблюдении ограничений $f_j(x) \leq 0$, $j = 1, \dots, m$. При этом предполагается заранее известным некоторое замкнутое подмножество G n -мерного векторного пространства, из которого следует выбирать решение x . Таким образом, задача математического программирования имеет вид

$$f_0(x) \rightarrow \min \mid f_j(x) \leq 0, \quad j = 1, \dots, m, \quad x \in G \subseteq R^n.$$

В зависимости от свойств функций $f = (f_0, f_1, \dots, f_m)$ и множества G имеют дело с тем или иным классом задач оптимизации. Известно огромное количество экономических и технических задач планирования, управления и проектирования, которые укладываются в приведенную схему математического программирования. Поэтому оценка трудоемкости численных методов оптимизации представляет не только теоретический, но и практический интерес.

2. Приведем формальное описание понятия «метод решения задач данного класса». Оно основано на так называемом информационном подходе, предложенном Н. С. Бахваловым в начале 60-х годов применительно к гораздо более общей ситуации.

Зафиксируем семейство (f, G) задач математического программирования и снабдим его источником информации об индивидуальных задачах класса — оракулом O . Оракул определяется множеством X допустимых вопросов, множеством Y возможных ответов и функцией наблюдения $\psi(x, f) : X \times (f, G) \rightarrow Y$, приводящей в соответствие каждому вопросу $x \in X$ по индивидуальной задаче f из рассматриваемого семейства ответ $y \in Y$.

Решение задачи численным методом B с оракулом O происходит итеративно по шагам. На каждом шаге метод «задает вопрос» $x \in X$ оракулу и получает от него ответ $y = \psi(x, f) \in Y$. Каждый следующий вопрос формируется по предыдущим вопросам и ответам. Методы различаются правилами выбора вопросов оракулу. На некотором шаге (по выбору метода) процедура решения останавливается и по накопленной к этому моменту информации метод формирует определяемое им приближенное решение z задачи f .

Метод B численного решения задачи представляет собой набор инструкций по формированию очередных вопросов, выбору момента остановки и построению приближенного решения в зависимости от накопленной к очередному шагу информации о решаемой задаче. Таким образом, метод одновременно «идентифицирует» индивидуальную задачу класса (в окрестности ее решения) и решает ее.

3. Метод B , использующий оракул O для решения задач семейства (f, G) , характеризуется своей трудоемкостью и погрешностью. Трудоемкостью метода B на задаче f класса называется число шагов, требуемое для решения задачи (число вопросов, задаваемых оракулу при решении задачи). Трудоемкостью метода B на классе задач называется верхняя грань трудоемкостей индивидуальных задач семейства. Погрешностью метода B на индивидуальной задаче f класса (т. е. погрешностью результата z метода B в качестве приближенного решения задачи f) называется величина

$$\varepsilon = \varepsilon(z, f) = \max \{f_0(z) - f_*, f_1(z), \dots, f_m(z)\},$$

где f_* — оптимальное значение целевой функции $f_0(x)$. Задача f математического программирования считается решенной с погрешностью, не превышающей ε , если найден вектор $z \in G$, для которого

$$f_0(z) - f_* \leq \varepsilon, f_j(z) \leq \varepsilon, j = 1, \dots, m.$$

Погрешность метода B на классе задач называется верхняя

грань по f погрешностей метода B на индивидуальных задачах класса.

Информационной сложностью $N(\epsilon)$ решения задач класса (f, G) с помощью методов B , использующих оракул O и обеспечивающих приближение с погрешностью, не большей ϵ , называется потенциальная граница эффективности таких методов, определяемая как минимальная трудоемкость, с которой метод B может еще решить любую индивидуальную задачу класса с погрешностью, не превышающей ϵ . Другими словами, информационная сложность $N(\epsilon)$ класса задач (f, G) — это минимальное число l шагов, при котором еще существует некоторый метод B , использующий оракул O и решающий задачи класса с трудоемкостью не более l и погрешностью не более ϵ .

В теории информационной сложности оцениваются сложности основных классов экстремальных задач и конструируются методы, оптимальные или почти оптимальные по сложности. Таким образом, теория информационной сложности — это теория построения и оценки экономных методов решения оптимизационных задач различных классов.

4. В заключение параграфа обсудим достоинства и недостатки принятого подхода к оценке классов задач оптимизации и методов их решения.

Главные достоинства подхода — его общность и результативность. В описанную схему укладывается любой мыслимый численный метод оптимизации. Оказывается, что даже в столь широкой постановке удастся получить содержательные результаты. Недостаток подхода — грубость измерения трудоемкости метода. Трудоемкость решения задач характеризуется здесь лишь объемом информации, которую нужно получить, чтобы построить решение заданной точности. При этом не учитывается реальный расход вычислительных ресурсов на переработку этой информации. Реальные затраты ресурсов могут существенно зависеть от выбранного метода и могут быть весьма велики. Грубость измерения трудоемкости — плата за общность и результативность подхода. Попытка уточнить определение трудоемкости, заменив, например, число шагов метода числом необходимых арифметических операций для столь обширных и аморфных по структуре классов задач, как задачи нелинейного программирования, привела бы к ряду малообозримых проблем, решение которых неминуемо содержало бы элементы произвола.

Грубым представляется и принятый минимаксный подход — оценка метода по результатам его применения к «наихудшим» задачам класса. Казалось бы, что гораздо более объективными показателями метода были бы характеристики (трудоемкость и погрешность), усредненные по задачам класса. Однако для сколь-нибудь широких классов задач нет никаких содержательных и математически осмысленных способов задать априорное распределение задач в классе. Это значит, что минимаксная постановка представляет единственной доставляющей объективную информацию. Тем не менее мы увидим, что и при таком грубом подходе к оценке сложности задач можно прийти к вполне осмысленным и важным результатам о потенциальных возможностях методов оптимизации.

Следует, правда, отметить, что в самое последнее время появились работы, заставляющие смягчить приведенные выводы. Можно ожидать, что при некоторых естественных требованиях к распределению задач в классах удастся получить оценки средней трудоемкости методов оптимизации и получить, таким образом, возможность более объективного их сравнения.

Информационная сложность нелинейного (невывуклого) программирования

1. Рассмотрим класс всех гладких невыпуклых задач нелинейного программирования

$$f(x) \rightarrow \min | x \in G,$$

где G — единичный шар в n -мерном пространстве R^n , а функция f на R^n дифференцируема сколь угодно много раз, причем ее частные производные до порядка k включительно во всем пространстве по модулю не превосходят 1. Пусть оракул сообщает значения производных f всех порядков (от 1 до k) в любой точке, в которой ему задается вопрос. Оказывается, информационная сложность такого класса задач фантастически велика и оценивается снизу формулой

$$N(\epsilon) \geq c(n, k) \left(\frac{1}{\epsilon} \right)^{\frac{n}{k}},$$

где $c(n, k)$ — некоторая константа, зависящая от n и k . На практике k обычно невелико, $k=1-2$ (старшие производные мы оценивать обычно не умеем), так что при немалых n и $\frac{1}{\epsilon}$ разработка универсального метода (годного для решения любой задачи класса) с приемлемыми затратами вычислительных ресурсов — безнадежная затея. (Заметим, что значение $c(n, k)$ даже по заниженным оценкам таково, что при $n=20$, $\epsilon=0,001$ имеем $N(\epsilon) \geq 10^{14}$.) Полученная оценка и сформулированный вывод справедливы и для методов случайного поиска.

Таким образом, теория информационной сложности говорит о бесперспективности поиска универсальных методов решения любых невыпуклых задач и рекомендует разработку специальных методов, учитывающих структуру и специфику более узких классов задач, включающих интересующую прикладника конкретную проблему. Чем уже этот класс, чем больше особенностей структуры задач может быть учтено, тем больше шансов на «изобретение» метода приемлемой трудоемкости.

2. Заметим, что катастрофический рост сложности $N(\epsilon)$ с ростом n и $\frac{1}{\epsilon}$ вызван не столько многоэкстремальностью задач рассматриваемого класса, сколько их невыпуклостью. Если сузить рассмотренный выше класс задач и оставить в нем только унимодальные задачи (задачи с единственной во всем пространстве критической точкой — нулем градиента), то сложность класса уменьшится несущественно и ее оценка снизу примет вид:

$$N_1(\epsilon) \geq c_1(n, k) \left(\frac{1}{\epsilon} \right)^{\frac{n-1}{k}},$$

что немногим лучше оценки снизу для общей задачи нелинейного программирования. Оптимизационные задачи не очень малой размерности, в которых целевой функционал имеет единственный экстремум и точки перегиба, уже крайне сложны для решения. Таким образом, одноэкстремальность не является еще тем свойством невыпуклых задач, которое позволяет существенно сократить трудоемкость их решения. Выделение подклассов нелинейных задач, для которых могут быть разработаны эффективные методы анализа, должно быть основано на других структурных свойствах функционалов, определяющих условия задачи.

Информационная сложность выпуклого программирования

1. Класс общих выпуклых задач является достаточно широким классом задач нелинейного программирования, в который укладываются многие прикладные проблемы. Этот класс задач характеризуется вполне приемлемой трудоемкостью решения.

Напомним, что в задачах выпуклого программирования область G — выпуклое множество, а критерий $f_0(x)$ качества решения и функции $f_j(x)$, $j=1, \dots, m$, определяющие ограничения задачи, — выпуклые функции. **Множество G выпукло**, если отрезок, соединяющий любые две точки G , целиком принадлежит этому множеству. **Функция $f(x)$ выпукла** на выпуклом множестве G , если хорда, соединяющая любые две точки графика $f(x)$, лежит не ниже соответствующей дуги графика (точек графика, отвечающих тем же значениям аргумента x). Выпуклые функции $f(x)$ на выпуклом множестве G обладают следующим свойством. Какова бы ни была точка x_0 , принадлежащая внутренности множества G (обозначается $x_0 \in \text{int}G$), всегда можно через нее провести по крайней мере одну гиперплоскость (если $f(x)$ гладкая, то только одну), которая разделит область G на две части, такие, что в одной из них гарантируется выполнение неравенства $f(x) \geq f(x_0)$. Разделяющая гиперплоскость определяется локальной информацией об $f(x)$ в точке x_0 . Для гладких функций это значение и градиент $f(x)$ в точке x_0 ($f(x_0)$ и $\nabla f(x)|_{x=x_0}$). Если в точке x_0 не существует градиента ($f(x)$ — негладкая функция), то его роль играет так называемый опорный функционал — любой вектор, определяющий направление гиперплоскости, проходящей через x_0 и не пересекающей график $f(x)$. Таким образом, если, например, одно из ограничений задачи математического программирования не удовлетворяется в точке x_0 (при допустимой погрешности ϵ), т. е. $f_j(x_0) > \epsilon$, то можно через точку x_0 провести гиперплоскость, которая рассечет G на две части, во всех точках одной из которых это ограничение не будет выполняться: $f_j(x) \geq f_j(x_0) > \epsilon$. Если требуется найти точку $x \in G$, в которой $f_0(x)$ достигает минимума, то информация о поведении $f_0(x)$ в точке x_0 позволяет провести через x_0 гиперплоскость, отсекающую от G часть, в которой минимум $f_0(x)$ не может быть достигнут. На этом свойстве основаны некоторые эффективные методы выпуклого программирования.

Будем рассматривать задачи выпуклого программирования, в которых G — выпуклое замкнутое ограниченное тело в n -мерном пространстве, а колебания выпуклых функций $f_0(x), f_1(x), \dots, f_m(x)$, определяющих условия задач, не превышают единицы, т. е. $\max_G f_j(x) - \min_G f_j(x) \leq 1$.

Гладкость этих функций отнюдь не предполагается. Будем считать, что оракул сообщает значения функций $f_j(x)$ и их градиента в любой точке x , указываемой методом. Если в некоторой точке функция $f_j(x)$ не имеет градиента, оракул сообщает вместо него какой-нибудь отличный от нуля опорный функционал в этой точке (градиент к подходящим образом сглаженной в этой точке $f_j(x)$).

Для класса общих выпуклых задач получены верхняя и нижняя границы асимптотики сложности по $\varepsilon \rightarrow 0$. Она задается следующими соотношениями:

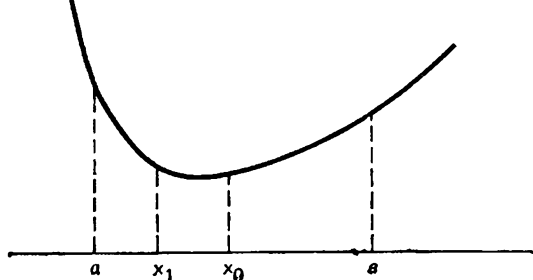
$$N(\varepsilon) \leq 2, 2 \left\lceil n \ln \frac{1}{\varepsilon} \right\rceil, \quad \varepsilon < 1,$$

$$N(\varepsilon) \geq \frac{1}{12} n \ln \frac{1}{\varepsilon}, \quad \varepsilon \leq \varepsilon(G), \quad \left(\varepsilon(G) \leq \frac{1}{n^2} \right).$$

(Здесь $\lceil a \rceil$ — наименьшее целое число, большее либо равное a ; $\varepsilon(G)$ — зависящая от G величина погрешности, начиная с которой справедлива нижняя оценка сложности.)

Таким образом, в принципе можно решать общие выпуклые задачи методом, сходящимся со скоростью геометрической прогрессии со знаменателем $1 - O\left(\frac{1}{n}\right)$. Этот метод не зависит от гладкости выпуклых функций $f_j(x)$ и от геометрических особенностей выпуклого тела G . В асимптотике по $\varepsilon \rightarrow 0$ быстрее решать общие выпуклые экстремальные задачи и нельзя.

2. Метод, реализующий при малых ε верхнюю оценку информационной сложности класса общих выпуклых задач, называется **методом центров тяжести (МЦТ)**. Он представляет собой естественный многомерный аналог метода минимизации одномерной выпуклой функции, известного под названием метода дихотомии, или метода деления пополам. Чтобы найти минимум одномерной функции $f(x)$ на отрезке $[a, b]$, находим на графике середину $[a, b]$ точку x_0 . Вправо от x_0 функция $f(x)$ растет, так что на отрезке $[x_0, b]$ $f(x)$ не может достигнуть минимума. Далее рассматриваем отрезок $[a, x_0]$, делим его точкой x_1 пополам и отбрасываем поло-



вину этого отрезка, в которой $f(x) \geq f(x_1)$. Продолжая этот процесс, мы можем с любой заранее заданной точностью определить минимальное значение целевой функции $f(x)$.

Обобщение метода дихотомии на многомерный случай — это метод центров тяжести. Уже указывалось, что, выбирая любую точку $x_0 \in \text{int}G$, можно по локальным характеристикам функций $f_j(x)$, отвечающим нарушенным в x_0 ограничениям, усекавать область $G_0 = G$, отбрасывая ее часть, заведомо не удовлетворяющую условиям задачи. Если в x_0 не нарушается ни одно из ограничений задачи, можно, используя локальные характеристики $f_0(x)$ в x_0 , отсечь часть области $G_0 = G$, в которой значения целевой функции заведомо превышают $f_0(x_0)$. Обозначим оставшуюся часть G через G_1 . Область G_1 также выпукла. Выберем в ней точку $x_1 \in \text{int}G_1$ и проведем через x_1 гиперплоскость, отсекающую часть G_1 , в которой не выполняются условия задачи, или (если все ограничения задачи в точке x_1 не нарушены) часть G_1 , в которой не может быть локализован минимум $f_0(x)$. Назовем оставшуюся часть G_1 через G_2 и продолжим процедуру. Таким образом, можно, последовательно выбирая точки $x_i \in \text{int}G_i$, сокращать объем области локализации решения задачи выпуклого программирования. Чтобы добиться достаточно быстрого убывания объемов частей G_i множества G , следует наиболее рациональным образом выбирать точки x_i . Наиболее эффективным оказывается выбор в качестве x_i центра тяжести G_i . В этом случае отношение объемов областей G_i и G_{i-1}

$$\frac{|G_i|}{|G_{i-1}|} \leq 1 - \left(\frac{n}{n+1} \right)^n \leq 1 - \frac{1}{e}.$$

Отвечающий этому правилу метод решения задач выпуклого программирования — метод центров тяжести — обеспечивает выполнение неравенства

$$\varepsilon(z_i, f_0) \leq \left(1 - \frac{1}{e}\right)^{\frac{i}{n}},$$

где z_i та из допустимых точек x_0, x_1, \dots, x_i , на которой достигается наименьшее значение $f_0(x)$. Из последнего неравенства следует приведенная ранее верхняя оценка сложности класса общих выпуклых задач.

3. Уже отмечалось, что оценка трудоемкости решения общих выпуклых задач, определяемая методом центров тяжести, правильно описывает поведение сложности в асимптотике по $\varepsilon \rightarrow 0$. При этом асимптотика устанавливается, начиная от некоторого $\varepsilon(G)$, зависящего от геометрических свойств G . Если G куб или параллелепипед, то $\varepsilon(G) = \frac{1}{8}$;

если G шар или эллипсоид, то $\varepsilon(G) = \frac{1}{n}$. Таким образом, при высоких требованиях к точности ($\varepsilon < \varepsilon(G)$) трудоемкость метода центров тяжести не может быть улучшена по порядку. К сожалению, этот результат имеет чисто теоретическое значение. Для реализации МЦТ необходимо уметь на каждом шаге отыскивать центр тяжести произвольных выпуклых тел. А это уже при $n > 3$ практически безнадежная в вычислительном отношении задача.

Однако ценой некоторого увеличения трудоемкости удается подправить идею метода центров тяжести и обеспечить приемлемую вычислительную сложность решения выпуклых задач. Соответствующая модификация алгоритма называется **модифицированным методом центров тяжести (ММЦТ)** или **методом эллипсоидов (МЭ)**.

Идея метода эллипсоидов заключается в том, что на каждом шаге область локализации решения вписывается (в подлежащей системе координат) в эллипсоид минимального объема, для которого вычисление центра тяжести не представляет труда. Оказывается, что для метода эллипсоидов

$$\frac{|G_i|}{|G_{i-1}|} \leq 1 - \frac{c(n)}{n},$$

где $0 < c(n) \xrightarrow{n \rightarrow \infty} \frac{1}{2}$. Отсюда следует, что оценка трудоем-

$$N(\varepsilon) \leq c \left\lceil n^2 \ln \frac{n^2}{\varepsilon} \right\rceil,$$

где c — некоторая константа.

С другой стороны, алгоритмическая реализация метода эллипсоидов относительно проста. Один шаг метода требует $O(n^2)$ арифметических операций при памяти $O(n^2)$.

Совсем недавно было показано, что на идее МЦТ может быть по аналогии с МЭ построен метод симплексов (МС), в котором роль эллипсоида на каждом шаге выполняет симплекс. Центр тяжести симплекса вычисляется без труда. Его координаты являются среднеарифметическими значениями координат вершин симплекса. Трудоемкость МС

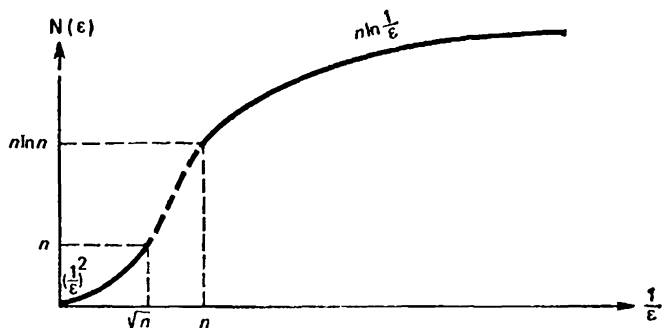
оценивается величиной $cn^3 \ln \frac{1}{\varepsilon}$. Имеются, однако, основания ожидать, что для широкого класса распределений средние оценки МС предпочтительней средних оценок МЭ.

Метод эллипсоидов и метод симплексов (МЭ и МС) являются единственными известными универсальными методами решения общих выпуклых задач с полиномиальной по n и $\frac{1}{\varepsilon}$ оценкой трудоемкости и полиномиальной по n вычислительной сложностью шага.

Накопленный опыт использования метода эллипсоидов для решения выпуклых задач умеренной размерности ($n=15 \div 25$) на современных вычислительных машинах показывает, что точность решения порядка $10^{-8} \div 10^{-9}$ достигается за $1 \div 15$ мин.

4. Мы видели, что асимптотика сложности класса общих выпуклых задач зависит только от размерности n тела G , а геометрические свойства G определяют момент установления асимптотики. Обычно выход на асимптотику происходит тем позже, чем больше n . В экономических и ряде других прикладных задач нередко n весьма велико ($10^3 \div 10^6$), а ε не слишком мало ($10^{-1} \div 10^{-2}$). Поэтому представляет интерес анализ поведения сложности (и реализующих ее методов) и в доасимптотическом диапазоне значений ε .

Оказывается, что в этом случае для тел G с «хорошими» геометрическими свойствами (например, для тел типа «шаров» в так называемых пространствах Лебега L_p , в частности для обычного шара в евклидовом пространстве L_2)



сложность ограничена сверху функцией, представляющей полином от $\frac{1}{\varepsilon}$ и не зависящей или почти не зависящей от размерности n . Приведенный график иллюстрирует изменение сложности класса общих выпуклых задач для случая, когда тело G хорошо аппроксимируется евклидовым шаром. В начальном — доасимптотическом — диапазоне изменения $\frac{1}{\varepsilon}$ сложность растет, как $\frac{1}{\varepsilon^2}$, и не зависит от размерности n . Так продолжается до тех пор, пока сложность не достигнет величины порядка размерности G . В конечном диапазоне значений $\frac{1}{\varepsilon}$ (при $\frac{1}{\varepsilon} \geq n^2$) сложность меняется, как $n \ln \frac{1}{\varepsilon}$. В промежуточном диапазоне порядок сложности меняется от n до $n \ln n$. Примерно так же ведет себя график сложности выпуклых задач в случаях, когда G аппроксимируется шаром в L_p , $1 < p < \infty$, только при малых $\frac{1}{\varepsilon}$

$$N(\varepsilon) \sim \left(\frac{1}{\varepsilon}\right)^{\max(2, p)};$$

$$\text{при } p=1 \quad N(\varepsilon) \sim \left(\frac{1}{\varepsilon}\right)^2 \ln n,$$

$$\text{при } p=\infty \quad N(\varepsilon) \sim n \ln \frac{1}{\varepsilon}.$$

В случаях когда G — евклидов шар, верхняя оценка сложности (в начальном диапазоне изменения $\frac{1}{\varepsilon}$) реали-

зуется обычным градиентным методом негладкой оптимизации. Если G шар в L_p , соответствующие оценки сложности реализуются так называемыми методами зеркального спуска [33]. Эти методы представляют не только теоретический интерес. Они позволяют приспособить метод к геометрическим особенностям решаемой задачи. В некоторых типичных задачах большой размерности (например, в характерных для ряда экономических и технических задач случаях, когда G представляет собой симплекс) применение методов зеркального спуска дает ощутимый эффект.

5. В теории информационной сложности выпуклых задач имеются и оценки трудоемкости, и эффективные методы решения для различных частных классов выпуклых задач, в частности, для классов гладких и сильно выпуклых задач. Методы, реализующие верхнюю оценку сложности, обобщены и приспособлены к решению выпукло-вогнутых игр, выпуклых задач управления при неполной информации, обобщенных выпуклых задач, в которых требуется из некоторого выпуклого множества выбрать вектор, оптимальный в смысле заданного выпуклого предпочтения. Обзор постановок и результатов в этих направлениях см. в [35]. Намечились также и некоторые другие приложения результатов теории информационной сложности.

Субоптимальные законы управления

1. В [55] обращено внимание на то, что методы оптимизации, субоптимальные по информационной сложности, в ряде случаев оказываются и субоптимальными «законами управления», т. е. методами, обеспечивающими оптимизацию в реальном времени.

Подчеркнем еще раз, что рассмотренные выше методы оптимизации выпуклых функций могут быть эффективно распространены на стохастический случай, когда информация о значениях функций и их градиентов (или опорных функционалов) искажена случайными ошибками, распределение которых удовлетворяет некоторым нежестким требованиям.

Решение условной экстремальной задачи можно интерпретировать в содержательных терминах как формирование управления реальным объектом. В момент t выбирается команда управления x_t , принадлежащая допустимому множеству управлений G и минимизирующая заданную функ-

цию потерь $f(x)$. До сих пор, говоря о решении задачи численным методом, мы не предполагали взаимодействия с реальным объектом. Информация о локальных характеристиках функции потерь в выбранной точке x_i не определялась реальной реакцией объекта на управляющее воздействие x_i . Стратегия выбора точек x_i , в которых задавались вопросы источнику информации, оценивалась не тем, насколько хороши точки x_i как команды управления, а тем, как быстро локальная информация о функции потерь в точках x_i окажется достаточной для построения решения требуемой точности. Такой подход оправдан при работе с моделью, а не с реальным объектом. На практике часто возникает необходимость в построении адаптивной системы управления, в которой формирование управляющих воздействий на априори не полностью описанный объект происходит в ходе работы с ним по результатам анализа его реакций на предыдущие воздействия. Оказывается, что несложная модификация методов зеркального спуска может обеспечить адаптивное управление объектом.

2. Пусть на объект управления в моменты $t=1, 2, \dots$ подаются команды $x_t \in G$. Будем считать, что в момент t на объект, кроме того, воздействует помеха ω_t . Предполагается, что реализации помехи в различные моменты времени — независимые, одинаково распределенные случайные величины.

Мы рассматриваем безынерционный объект, поведение которого в момент t полностью описывается командой x_t и реализацией помехи ω_t и не зависит от предыстории воздействий на объект. Поведение объекта в момент t оценивается функцией потерь $\varphi(x_t, \omega_t)$. После реализации команды управления x_t и помехи ω_t в орган управления по цепи обратной связи поступает значение функции потерь φ и вектор градиента $\nabla_x \varphi$ в этой точке. Эта информация используется для формирования очередного управляющего воздействия x_{t+1} . Правила построения очередной команды как функции от накопленной к этому времени информации

$$\{\varphi(x_1, \omega_1), \nabla_x \varphi(x_1, \omega_1), \dots, \varphi(x_t, \omega_t), \nabla_x \varphi(x_t, \omega_t)\}$$

и представляют закон управления объектом.

Будем характеризовать качество управления средним по времени и по реализациям помех значением функции потерь

$$g(t) = M_{\omega} \left\{ \frac{1}{t} \sum_{\tau=1}^t \varphi(x_{\tau}, \omega_{\tau}) \right\}.$$

Оптимальный закон управления обеспечивает минимизацию $g(t)$.

Будем считать, что область определения управляющих воздействий G выпукла, замкнута и ограничена, а функция $\varphi(x, \omega)$ выпукла (и липшицева по $x \in G$).

Обозначим $f(x) = M_{\omega} \varphi(x, \omega)$ и пусть x^* есть точка, в которой $f(x)$ достигает своего минимального значения на G . Поскольку x_t формируется по предыдущим командам и, следовательно, зависит от истории процесса до момента t , а ω_t не зависит от предшествующих помех $\omega_1, \dots, \omega_{t-1}$, то $g(t) \geq f(x^*)$. С другой стороны, при $x_t \equiv x^*$ $g(t) = f(x^*)$ при всех t . Однако если объект управления заранее полностью не определен, то управление $x_t = x^*$ не может быть реализовано никаким законом управления и нужно заботиться лишь о том, чтобы $g(t)$ возможно быстрее приблизилось к $f(x^*)$.

Выбор субоптимального численного метода решения стохастической задачи выпуклой оптимизации и формирование субоптимального закона управления — близкие проблемы, различающиеся тем, что рациональный численный метод оценивается по числу шагов, затрачиваемых на достижение решения требуемого качества, а закон управления оценивается по средним потерям, определяемым использованием формируемых в ходе поиска управлений.

3. Обозначим через $\varepsilon_{\tau} = M_{\omega} \varphi(x_{\tau}, \omega_{\tau}) - f(x^*)$ среднее значение погрешности τ -й команды x_{τ} , обусловленное выбранным методом оптимизации, в качестве решения экстремальной задачи

$$f(x) \rightarrow \min \text{ при } x \in G.$$

Средние (по времени и помехам) потери $g(t)$ превышают минимально возможное значение на величину

$$\Delta(t) = g(t) - f(x^*) = \frac{1}{t} \sum_{\tau=1}^t \varepsilon_{\tau}.$$

В силу выпуклости $f(x)$ имеет место неравенство

$$\begin{aligned} \frac{1}{t} \sum_1^t \varepsilon_{\tau} & \left(= \frac{1}{t} \sum_1^t \{M_{\omega_{\tau}} \varphi(x_{\tau}, \omega_{\tau}) - f(x^*)\} \right) = \\ & = \frac{1}{t} \sum_1^t (f(x_{\tau}) - f(x^*)) \geq f\left(\frac{1}{t} \sum_1^t x_{\tau}\right) - f(x^*). \end{aligned}$$

Пусть погрешность $\Delta(t)$ метода оптимизации (который в рассматриваемом случае используется в качестве закона управления) в момент t не превышает требуемой величины ϵ . Если считать результатом работы метода в момент t точку $\frac{1}{t} \sum_{\tau=1}^t x_{\tau}$, то трудоемкость такого метода равна t , а погрешность не превышает $\Delta(t) \leq \epsilon$. Число шагов, гарантирующее решение задачи оптимизации с погрешностью ϵ , не больше минимально достижимого числа шагов, после которых «наилучший» закон управления обеспечит выполнение неравенства $\Delta(t) \leq \epsilon$. Обратное, вообще говоря, неверно. Если некоторый метод оптимизации позволяет находить решение погрешности ϵ за t шагов, то это еще не значит, что он в качестве закона управления в состоянии гарантировать при всех $T \geq t$ неравенство $\Delta(T) \leq \epsilon$. Тем не менее существуют методы, «субоптимальные» в качестве вычислительных методов выпуклой оптимизации и в то же время «субоптимальные» в качестве закона управления. Такими методами, в частности, являются упомянутые выше методы зеркального спуска. Соответствующее обсуждение и аргументация приведены в [55].

Адаптивное управление экономикой

1. В [34] рассмотрено еще одно приложение теории информационно-сложности на этот раз к теоретическим аспектам адаптивного управления экономикой.

В известных экономико-математических моделях управления экономической системой обычно предполагается возможность прогнозирования результатов реализации любых допустимых воздействий. В этом предположении выбираются оптимальные в том или ином смысле управления и исследуются их свойства.

Предположение об априорном знании управляемой системы допустимо, если исследованию подлежат «локальные» проблемы, связанные с выбором и анализом управляющих воздействий «вблизи» уже реализованных. Это предположение, однако, уязвимо, если речь идет о «глобальном» управлении экономикой на больших временных интервалах.

В настоящем параграфе рассматривается ситуация, в которой требуется управлять экономикой с априори не полностью известными последствиями принятых управлен-

ческих решений. Предполагается, что заранее известны лишь некоторые качественные характеристики системы, так что априорная информация выделяет не одну конкретную экономическую систему, а достаточно широкий их класс. При этом заранее неизвестно, с какой именно системой из этого класса придется иметь дело. Всю недостающую информацию можно получить лишь в ходе реального управления, цель которого предполагается известной. Методология теории информационной сложности позволяет при некоторых естественных предположениях о классе систем построить эффективный (в некотором смысле субоптимальный) механизм накопления информации и управления, обеспечивающий достижение поставленной цели.

При неполной априорной информации об управляющей системе качество механизма управления зависит не только от него, но и от конкретной системы, к которой он применяется. В таких случаях естественно сравнивать различные способы управления по определяемым ими гарантиям. Другими словами, будем считать лучшим механизмом управления механизм, оптимальный в минимаксном смысле. Такой механизм обеспечивает возможно лучшее управление в наихудшей ситуации.

В отличие от известных экономико-математических моделей, в которых цель управления — максимизация темпов роста экономики, предлагаемый подход преследует другую цель — стабилизацию экономики в состоянии с наивысшей полезностью за возможно меньшее время. Аргументация этого подхода остается за рамками настоящей работы.

Опишем класс рассматриваемых экономических систем — определим основные понятия, сформулируем цели управления и обсудим априорные гипотезы о системах, гарантирующие достижение поставленных целей.

2. Начнем с описания экономической системы.

Под **экономической системой** будем понимать совокупность трех объектов: производственной системы, дисциплины распределения наличного продукта и функции полезности.

(а) **Производственная система.** Будем рассматривать экономическую систему, в которой обращаются n видов продукции. Функционирование системы происходит по тактам (например, годам). Продукция, произведенная в данном году, определяется планом этого года. Ясно, что возможности управления не ограничиваются определением вектора

продуктов x , предназначенного на данном этапе для производственного использования. Помимо этого, можно еще давать не стесненные наличными материальными ресурсами директивные указания. План z не только фиксирует выделенные производству материальные и трудовые ресурсы x , но и определяет директивные задания u по распределению этих ресурсов между отраслями. Таким образом, план z характеризуется двумя группами компонент x и u $z = (x, u)$. Вектор $x = (x^1, \dots, x^n) \in R_x^n$ — это вектор продуктов, предназначенных для производственного использования в данном году. Вектор $u = (u^1, \dots, u^m) \in R_u^m$ — это вектор плановых директив — команд производству со стороны управляющего органа. Имеем $z = (x, u) \in R_x^n \times R_u^m = R_z^{n+m}$. Будем обозначать x -компоненту плана z через $P(z)$.

Под **производственной функцией** экономики $T(z) = (T_1(z), \dots, T_n(z))$ будем понимать вектор-функцию, определяющую производство разных видов продукции в результате реализации плана z .

Планы, которые могут быть в принципе реализованы, образуют некоторое множество $G \subset R_z^{n+m}$. Естественно принять, что G ограничено, а $P(G)$ лежит в неотрицательном ортанте R_x^n .

Будем называть пару (G, T) **производственной системой**.

(б) Распределение ресурсов. Продукт, имеющийся в системе, на каждом такте распределяется между непродовственным потреблением на данном такте и производственными затратами на следующем такте и, возможно, частично переводится в запас. Дисциплина распределения определяет на каждом такте t объем продукта $R_t \in R_x^n$, отведенный для производственного использования на такте t , как функцию от «истории» системы до момента t . При этом R_1 — начальное условие — предполагается заданной величиной.

Допустимый план на такте t — это план, удовлетворяющий условиям $z_t \in G$, $P(z_t) \leq R_t$, т. е. план, реализуемый технически и обеспеченный ресурсами.

Таким образом, функционирование производства определяется траекторией планов z_1, z_2, \dots , которой соответствует траектория объемов производимых продуктов $T(z_1), T(z_2), \dots$. Последняя, в свою очередь, определяет в соответствии с дисциплиной распределения наличного продукта траекторию R_1, R_2, \dots , объемов продукта, расходуемого на производство.

(в) Полезность. Предполагается, что удовлетворение от

потребления в непроеизводственной сфере вектора продуктов x характеризуется функцией полезности $V(x)$.

3. Цели управления. Будем считать, что целью управления является возможно более быстрая стабилизация экономики в состоянии с максимальной полезностью.

Чтобы формализовать и, таким образом, уточнить цель управления, введем понятие стационарного состояния (или стационарного плана) экономики. Назовем **стационарным планом** экономики любой план $z \in G$, для которого $T(z) \geq P(z)$. Если z стационарный план и в момент t вектор ресурсов R_t допускает реализацию плана z (т. е. $R_t \geq P(z)$), то, начиная с момента t , можно все время применять план z , изымая для непроеизводственного потребления «излишек» $T(z) - P(z)$ (чтобы исключить ситуации, в которых «излишка» может не оказаться, будем считать, что под $T(z)$ подразумевается производственная функция, из которой заранее исключен необходимый минимум непроеизводственного потребления). Таким образом, можно стабилизировать экономику в соответствии с планом z и полезностью $V^*(z) = V(T(z) - P(z))$, называемой **стационарной полезностью**.

Во введенных терминах цель управления состоит в наивысшем переводе экономики в стационарное состояние с максимально возможной стационарной полезностью $V^* = \max\{V^*(z) | z \in G \text{ — стационарно}\}$. Это состояние будем называть оптимальным. Можно показать, что при естественных предположениях относительно характеристик экономики стабилизация в оптимальном состоянии максимизирует среднюю по времени полезность управления.

4. Априорные гипотезы. Сформулируем предположения об управляемой экономической системе, гарантирующие достижение поставленной цели

Помимо общепринятых допущений о выпуклости, замкнутости и ограниченности G и вогнутости компонент производственной функции и функции полезности, мы будем предполагать экономику восстанавливаемой и расширяемой. Определим эти понятия.

Экономика называется **восстанавливаемой**, если существует (и известен) по крайней мере один план $\tilde{z} \in G$, обладающий следующими свойствами

(1) \tilde{z} — «продуктивный» план, т. е. $T(\tilde{z}) > P(\tilde{z})$ (неравенство по каждой компоненте);

(2) какова бы ни была реализуемая траектория разви-

тия системы до такта t , вектор R_t позволяет реализовать в такте t план \tilde{z} .

План \tilde{z} , удовлетворяющий требованиям (1) и (2), называется **точкой восстановления экономики**.

Условие восстанавливаемости означает, что, несмотря на то что экономическая система не предполагается априори заданной, существует гарантия от попадания системы в безвыходное состояние — существует «продуктивный» план, достижимый за один шаг из любого другого состояния.

Пусть $\lambda \in R^n$ $\lambda = (\lambda^1, \dots, \lambda^n) > 0$. Экономика называется **λ -расширяемой**, если при реализации в произвольный момент t произвольного стационарного плана $z_t = (x_t, u_t)$ выполняется условие $R_{t+1}^j \geq x_t^j + \lambda^j (T_j(z_t) - x_t^j)$, $j = 1, \dots, n$. Другими словами, экономика λ -расширяема, если дисциплина распределения имеющегося продукта такова, что, получив по j -му продукту прирост выпуска над затратами $T_j(z) - x^j$, можно расширить вложение в производство продукта не менее чем на долю λ^j от полученного прироста.

5. Будем предполагать, что после реализации плана z_t управляющему органу становятся известными вектор $T(z_t)$ и матрица $T'(z)$ субградиентов компонент производственной функции в точке t . Эта локальная информация используется для формирования очередного плана.

Обозначим через A множество всех экономических систем, удовлетворяющих перечисленным в предыдущих пунктах допущениям.

Приведем идею механизма управления — набора правил формирования очередных планов в зависимости от накопленной к текущему такту информации. От механизма управления требуется, чтобы на любой системе из A он определял траекторию планов, стабилизирующуюся на ε -оптимальном по полезности стационарном плане. Качество механизма управления конкретной системой будем оценивать по времени достижения ε -оптимального стационарного плана. Чем оно меньше, тем механизм лучше.

Требуемый механизм управления должен в конечном счете обеспечить решение системы выпуклых неравенств:

$$z \in G: T(z) \geq P(z), \quad \bar{V}(z) \geq (1 - \varepsilon) \bar{V},$$

где

$$\bar{V}(z) = V(T(z) - P(z)), \quad \bar{V} = \sup \{ \bar{V}(z) | z \in G, T(z) \geq P(z) \}.$$

Первое из этих неравенств есть условие стационарности плана. Второе неравенство представляет собой условие ϵ -оптимальности плана. При этом требуется выйти на траекторию, удовлетворяющую этим неравенствам, за возможно меньшее время.

В [34] предложен механизм, реализующий решение сформулированной задачи. Механизм управления существенно использует схему метода центров тяжести.

Содержательно механизм управления вначале переводит систему из начального состояния в состояние с максимальным темпом роста, а уже из него переводит систему в ϵ -оптимальное по полезности состояние. Другими словами, вначале, абстрагируясь от полезности, определяется стационарный план с наибольшим темпом роста — строится материально-техническая база дальнейшего движения. Далее, отправляясь от этого плана, оптимизируется полезность.

Каждый такт первого этапа позволяет сократить область, «подозрительную на высокий показатель роста». Каждый такт второго этапа сокращает область, «подозрительную на высокую полезность».

6. Приведем оценку числа тактов $\tau(\mathcal{E}, \epsilon)$, за которое экономическая система \mathcal{E} , управляемая предложенным в [34] механизмом, гарантированно перейдет в ϵ -оптимальное состояние.

Выражение «экономика \mathcal{E} , спустя $\tau(\mathcal{E}, \epsilon)$ тактов, гарантированно застabilизируется в ϵ -оптимальном состоянии» означает, что накопленная за $\tau(\mathcal{E}, \epsilon)$ тактов информация позволяет сделать вывод о том, что достигнутое состояние нельзя существенно улучшить и, стало быть, это состояние имеет смысл (и возможно) «заморозить» на все последующее время.

Назовем темпом роста экономики \mathcal{E} на плане z величину

$$\psi(z) = \min_{1 \leq j \leq n} \lambda_j'(T_j(z) - P_j'(z))$$

— часть приращения выпуска над затратами, которую допускается использовать для расширения производства наиболее критичного продукта.

Максимум темпа роста экономики \mathcal{E} по всем допустимым планам $z \in G$ обозначим через $a(\mathcal{E})$, а темп роста в точке \tilde{z} восстановления экономики — через $\alpha(\mathcal{E})$. Величина $Q = \frac{a(\mathcal{E})}{\alpha(\mathcal{E})}$ характеризует уровень потенциальных потерь, связанных

с переходом в точку восстановления \tilde{z} . Это значит, что существует стационарный план с темпом роста в Q раз большим, чем темп роста в \tilde{z} .

Пусть $\bar{x}' = \max_{z \in G} \{P^j(z) | z \in G\}$ максимальный по допустимым планам объем продукта x' , предназначенный для производственного использования, а $\mu(\Theta) = \max_{1 \leq i \leq n} \lambda' \bar{x}'$. Вели-

чина $S = \frac{a(\Theta)}{\mu(\Theta)}$ характеризует, грубо говоря, отношение максимально возможного прироста продукции в ходе производства к максимально возможному объему продукции.

Пусть $S < \frac{1}{2}$, $\varepsilon < \frac{1}{2}$, $a(\Theta) < 2$, $Q > 2$. В этих предположениях имеет место следующая оценка числа тактов $\tau(\Theta, \varepsilon)$, при котором предложенный в [34] механизм управления гарантирует стабилизацию экономики Θ в ε -оптимальном состоянии:

$$\tau(\Theta, \varepsilon) \leq \sigma \frac{m+n}{a(\Theta)} \left\{ \ln Q \ln \frac{1}{\varepsilon} - \ln Q \ln S + \ln^2 \frac{1}{\varepsilon} \right\}.$$

Здесь σ — абсолютная константа (по грубой оценке порядка не более 10^2). Таким образом, с точностью до логарифмических по Q , ε и S членов время стабилизации обратно пропорционально темпу роста экономики.

В [34] показано, что такое же неравенство может быть использовано для оценки эффективности механизма управления экономикой и в случае, когда производство продукции определяется не только реализованным планом, но и некоторыми непредсказуемыми факторами, определяющими «состояние среды» (погода, конъюнктура и т. д.). Правда, при этом речь идет уже не о стабилизации системы в состоянии с максимальной полезностью, а о стабилизации в состоянии с заданной (достижимой) полезностью V_0 . Кроме того, в этом случае качество механизма управления определяется числом тактов, на которых желательный уровень полезности не достигается (чем оно меньше, тем лучше).

Диалоговое программирование

1. Экономные алгоритмы решения задач выпуклой оптимизации, рассмотренные в этой главе, могут быть использованы также для организации рационального диалога между специалистами по содержательным и формальным

дисциплинам в процессе постановки и анализа задач планирования, управления и проектирования.

Одна из наиболее серьезных причин недостаточной эффективности разработок и внедрения многих серьезных идей и проектов, в создании которых принимали участие большие коллективы ученых, заключается в неудовлетворительной организации взаимодействия специалистов различного профиля, в первую очередь математиков и инженеров, математиков и экономистов.

Математики — сторонники формального мышления — нередко вынуждают инженеров, экономистов и представителей других прикладных дисциплин, мыслящих в содержательных терминах, осваивать несвойственные их характеру и профессиональной подготовке формальные конструкции и непривычные для них логические схемы рассуждений. В свою очередь прикладники, ограниченные сроками выполнения работ, требуют от математиков, участвующих в постановке и решении прикладных задач, отказа от строгих формальных построений и использования эвристических приемов, невзирая на отсутствие у них интуиции в данной области и достаточного практического опыта. В итоге при такой организации работы неэффективно используются квалификация и возможности и тех и других.

Среди современных прикладных проблем, требующих количественного анализа, редко встречаются задачи, исследование которых можно было бы четко разделить на два этапа — содержательный (этап постановки задачи и накопления исходной информации) и формально-математический (этап разработки метода анализа и формального решения задачи). Гораздо чаще этапы постановки и формального исследования задачи сложным образом переплетаются.

К моменту, когда осознается необходимость в разработке некоторой системы и принимается решение о формировании соответствующих научных коллективов и финансировании работы, обычно нет еще четко поставленной задачи. Начальная содержательная постановка задачи обычно допускает далеко не однозначное истолкование. Цели системы и ограничения на ее создание выделяют на первом этапе разработки не индивидуальную, как это понимают математики, задачу, а целый класс задач — массовую задачу. Априорная информация о массовой задаче, отвечающей исследуемой проблеме, определяется содержательными соображениями о структуре системы и характеристиках ее функционирования.

Четкая содержательная постановка задачи — необходимое, но отнюдь не достаточное условие для построения формальной модели, поддающейся количественному анализу. Не всегда удается получить аналитическую или другую формальную запись зависимостей, определяющих условия задачи. Это усложняет организацию взаимодействия специалистов различного профиля при постановке и решении задач синтеза сложных систем.

2. Новые подходы к построению субоптимальных методов оптимизации, основанные на теории информационной сложности, указывают пути рационального разделения труда и ответственности между математиками и вычислителями, с одной стороны, и инженерами и экономистами — с другой. Аналогичные формальные проблемы возникают при согласовании интересов заказчика и исполнителей, руководителей работы различного уровня и т. д. Во всех этих случаях процесс постановки и решения задачи представляет собой целенаправленный диалог между лицом, ответственным за разработку рациональной последовательности этапов накопления информации, и представителями содержательных дисциплин, обеспечивающих этап информацией.

Будем называть методологию постановки и решения задач управления, планирования и проектирования при ограниченной четко зафиксированной априорной информации об условиях задачи **диалоговым (полилоговым) программированием**. Термин «диалоговое программирование» часто используется как наименование различных интерактивных процедур взаимодействия оператора с ЭВМ при отладке программ или выборе решений. Последующий текст не оставляет, однако, места для разночтений.

Диалоговое программирование (ДП) позволяет посредством экономно организованного диалога между участниками разработки свести задачу **синтеза** системы к последовательности задач **анализа** некоторой совокупности систем. При заданных характеристиках источника информации (оракула) диалог считается рационально организованным, если идентификация задачи и вычисление решения требуемого качества проводятся наиболее экономным образом — при возможно меньшем числе вопросов математика и ответов оракула. В диалоговом программировании решение «наиболее сложной» индивидуальной задачи, принадлежащей **массовой задаче**, обусловленной исходной априорной информацией, проводится за минимально возможное число элементарных актов диалога.

3. Будучи математической теорией, диалоговое программирование, естественно, ограничивается рассмотрением формализуемых задач выбора решений. Это задачи математического программирования вида

$$P_0: f_0(x) \rightarrow \min | f_j(x) \leq 0, j = 1, \dots, m; x \in G \subseteq R^n.$$

К задачам вида (P_0) сводятся (по крайней мере в принципе) многие проблемы, связанные с выбором решения, однако, как уже указывалось, эта сводимость в сколь-нибудь сложных ситуациях есть в действительности сводимость лишь «в принципе».

Обычно мы не можем рассчитывать на знание всех функций, входящих в условия задачи. Априорная информация об условиях выбора решения не позволяет, как правило, построить достаточно полную формальную модель, отвечающую исследуемой содержательной задаче. Конечно, можно было бы попытаться восполнить пробелы в наших знаниях соответствующими исследованиями. Однако в сложных ситуациях объем недостающей информации так велик, а получение ее столь трудоемко, что такого рода подход нереализуем. Кроме того, он порочен и в методическом отношении. Для отыскания решения не требуется знать о задаче все. Необходима информация гораздо меньшего объема, но «нужного характера». Если же мы попытаемся получить информацию, полностью идентифицирующую задачу, то наряду с указанной необходимой информацией мы должны узнать много объективно лишнего.

Диалоговое программирование как раз и изучает вопрос о том, как наилучшим образом организовать процесс накопления информации о решаемой задаче, чтобы возможно быстрее накопить минимальный объем знаний, по которому уже можно принять решение требуемого качества.

4. Поясним сущность диалогового программирования на следующей интерпретации задачи P_0 . Пусть эта задача отвечает формированию некоторой технической системы. Решение x определяет конкретную ее реализацию. Множество G описывает область допустимых систем. Функционалы $f_0(x), \dots, f_m(x)$ выражают характеристики системы (затраты ресурсов, время разработки, функционально-технические характеристики и т. д.). Одна из этих характеристик ($f_0(x)$) подлежит оптимизации, тогда как на другие наложены определенные требования, ограничивающие допустимые диапазоны их изменения. Мы имеем, таким образом, дело с проблемой оптимального синтеза технической системы.

Задачи такого рода весьма сложны. Гораздо более простыми являются задачи анализа. Задача анализа состоит в ответе на вопрос: каковы будут характеристики $f_j(x)$, $j=0, 1, \dots, m$, системы, отвечающей данному конкретному значению x ? Вопросы такого рода естественны для инженера. Для подготовки ответа на них он обладает широким диапазоном возможностей (от использования прошлого опыта и прямого эксперимента до теоретического анализа проблемы).

Ясно, что ответ на подобный вопрос содержит некоторую информацию о решаемой задаче. Диалоговое программирование (ДП), грубо говоря, исследует проблему наиболее эффективного использования информации, получаемой при решении последовательности задач анализа, для решения исходной задачи синтеза.

Процесс применения ДП выглядит так. Для «запуска» диалога необходима некоторая априорная информация — задание множества в принципе возможных решений и описание (грубое, качественное) свойств функционалов $f_j(x)$. После этого специалисту по содержательной стороне дела (инженеру) организатором диалога (лицом, принимающим решение, — ЛПР) ставится серия задач анализа, связанная с решениями $x_1, x_2, \dots, x_m \in G$. Формирование очередного анализируемого решения проводится ЛПР (с помощью математика) на основе всей полученной к данному моменту информации. После некоторого числа «вопросов к инженеру» ЛПР формирует результирующее решение x . Применяемая им методика гарантирует оптимальность этого решения (с заранее предписанной точностью).

В формально математическом отношении содержание ДП состоит в построении оптимальных дисциплин диалога для стандартных классов задач типа P_0 . «Оптимальность» здесь означает получение результата при минимальном числе «вопросов к инженеру». Для основных подклассов задач выпуклого программирования такие дисциплины построены (см. [55]). Это позволяет ставить вопрос практического применения намеченного подхода.

Подчеркнем «организационные» аспекты ДП. В диалоге «математик — инженер» каждая из сторон «остаётся на родной почве», говорит на своем «языке» — действует естественными для нее методами. При использовании ДП отпадает необходимость построения адекватной действительности формальной модели ситуации. Применение ДП расширяет тем самым круг содержательных задач, доступных для точных методов решения.

5. Источник информации — оракул — при его рациональном использовании обеспечивает в диалоговом программировании идентификацию задачи до уровня, при котором может быть получено решение требуемого качества. Ответы источника информации (оракула) на вопросы организатора диалога могут содержать, например, значения функционалов $f_j(x)$, определяющих цели и ограничения для указанной в вопросе системы x . Такой источник информации будем называть оракулом нулевого порядка. Более квалифицированный оракул (оракул первого порядка) может, помимо того, сообщить и градиенты соответствующих функционалов в точке, отвечающей той же системе. Оракул может отвечать на вопрос, реализуема заданная система или нет. Более опытный оракул способен, кроме того, назвать причину, по которой система x нереализуема, или указать «ближайшую» к ней (в некотором заранее обусловленном смысле) реализуемую систему. Ответы оракула могут быть точные или приближенные, детерминированные или искаженные случайными помехами. Оракул может быть стационарным или нестационарным, т. е. списки дозволенных вопросов могут сохраняться или меняться во времени. Естественно, что рациональная организация диалога требует предварительной информации о способностях оракула.

Подчеркнем, что математические аспекты диалогового программирования разработаны в настоящее время главным образом применительно к оракулам нулевого и особенно первого порядка (как точных, так и подверженных влиянию независимых от шага к шагу и централизованных помех).

Оракулы в диалоге (или полилоге) — это инженеры, экономисты или эксперты, обладающие опытом и неформальными содержательными знаниями в исследуемой проблеме и оснащенные необходимым оборудованием для моделирования и экспериментирования. Другой участник диалога, в некотором смысле его организатор — это ЛПР, которому помогает математик. При наличии необходимого технического обеспечения (операционной системы для стандартных классов задач проектирования и управления) можно в тех ситуациях, где это целесообразно, исключить математика из непосредственного участия в диалоге. В таких ситуациях диалоговое программирование обеспечивает рациональное разделение обязанностей между специалистами, обладающими наибольшей информацией в обсуждаемом круге вопросов. Следует ожидать, что таким образом можно построить рациональные процедуры планирования экспери-

мента, взаимодействия заказчика и исполнителя, врача и пациента. По-видимому, в этом же направлении будет совершенствоваться взаимодействие конструктора и ЭВМ — диалоговое программирование в узком смысле слова.

6. В принципе применение диалогового программирования возможно и целесообразно в процессах синтеза произвольных технических и других систем. Однако от принципиальной возможности до реального использования весьма большая дистанция. Главные трудности, которые здесь встречаются, — это проблемы, встающие перед специалистами по содержательной стороне дела, проблемы, связанные с построением ответов на вопросы организаторов диалога. От инженера требуется умение достаточно точно и достоверно решать задачи анализа. Время решения этих задач главным образом и определяет продолжительность диалога. Таким образом, главная задача, возникающая при применении ДП, состоит в создании комплекса средств, помогающих инженеру надежно и быстро решать задачи анализа. Таким комплексом средств лучше всего может служить система имитационного моделирования.

Еще одно условие применимости диалогового программирования — единство интересов участников диалога. Оракул может ошибаться (разработаны процедуры ДП со стохастическим оракулом), но не должен сознательно вводить организатора диалога (ЛПР) в заблуждение. Рассмотрим некоторые пути согласования интересов участников диалога. При этом одновременно сокращается и размерность задачи. Для определенности здесь рассматриваются лишь точные детерминированные оракулы нулевого порядка. Один из возможных подходов к постановке и решению задачи P_0 при несовпадении интересов участников решения задачи состоит в управлении оракулом O (оракулами O_i в случае полилога).

Пусть выбор решения зависит от s лиц O_i , а ЛПР — организатор L полилога — имеет возможность воздействовать на участников O_i выбора решения, обещая предоставить O_i вектор ресурсов $y_i \in G^{y_i} \subset R_i$. Задание векторов y_1, \dots, y_s (или, что то же самое, вектора $y^s = (y_1, \dots, y_s) \in G^y = (G^{y_1} \times \dots \times G^{y_s}) \subset R^s = (R_1, \dots, R_s)$) обуславливает решение $x = x(y^s)$, принятое группой O_1, \dots, O_s . Тем самым вектор y^s определяет числа $g_j(y^s) = f_j(x(y^s))$, $j = 1, \dots, m$.

Цель организатора L полилога — добиться принятия

группой O_1, \dots, O_s решения \bar{x} — оптимального плана задачи P_0 . Вообще говоря, эта цель недостижима, поскольку \mathcal{L} не контролирует сами решения — управление x не находится непосредственно в его руках. Самое большое, что может сделать \mathcal{L} , это решить задачу

$$P_1: g_0(y^s) \rightarrow \min | g_j(y^s) \leq 0, \quad j = 1, \dots, m; \quad y^s \in G^y.$$

На каждом шаге диалоговой процедуры решения этой задачи организатор полилога \mathcal{L} сообщает $O_i, i = 1, \dots, s$, вектор ресурсов $y_i \in G^{y_i} (y^s \in G^y \subset R^s)$, который может быть ему предоставлен для того, чтобы обеспечить решение x . В ответ \mathcal{L} получает от оракулов информацию $g_j(y^s) = f_j(x(y^s))$ о характеристиках решения $x = x(y^s)$, отвечающего вектору ресурсов y^s . Эта процедура может повторяться. В любой момент \mathcal{L} может прекратить торг и реализовать произвольное предложение на $G^y \subset \tilde{G}^y$ (отличие \tilde{G}^y от G^y естественно — обещать можно больше, чем имеешь).

Ясно, что работа оракула в задаче $P_1 \equiv \{g\}$ более трудоемка, чем в исходной задаче $P_0 \equiv \{f\}$. В задаче $\{f\}$ оракулу указывается решение x и требуется установить его характеристики — значения $f_j(x), j = 0, 1, \dots, m$, и, возможно, их опорные функционалы в точке x . В задаче $\{g\}$ от оракула требуется принять решение $x = x(y)$ (построить план x или спроектировать схему x), обеспеченные обещанными ему ресурсами y , и установить характеристики $f_j(x(y)) = g_j(y)$ (и, возможно, $\nabla_y g_j(y)$) этого решения. Однако для организатора \mathcal{L} полилога задача $\{g\}$, как правило, проще $\{f\}$: ее размерность обычно существенно ниже. Решая задачу $\{g\}$ диалоговыми методами, приближаются в той мере, в которой это допускается структурой и особенностями оракула, к решению «объективно существующей задачи $\{f\}$ ».

Возможны и другие подходы к управлению оракулом. Пусть, например, организатору диалога известна «цена ресурсов» — функция $h(y^s)$, заданная или допускающая диалоговое наблюдение. Тогда управление оракулом сводится к итеративному решению задачи

$$P_2: h(y^s) + g_0(y^s) \rightarrow \min | g_j(y^s) \leq 0, \quad j = 1, \dots, m; \quad y^s \in G^y.$$

7. Во многих ситуациях конкурентный механизм вынуждает исполнителей согласовывать свои интересы с интересами заказчика.

Пусть имеется s потенциальных подрядчиков, способных разрабатывать заданную систему. Заказчик (в данном случае организатор Л полилога) располагает суммой Y и стремится извлечь от реализации заказа максимальную прибыль. Обозначим через y_i сумму, которую заказчик на очередном шаге полилога предлагает i -му подрядчику. Исполнители в ответ сообщают векторы характеристик $x_i(y_i)$, $i=1, \dots, s$ заказываемой системы, которые каждый из них способен обеспечить за предложенную плату. Организатор Л полилога (или специальный эксперт) по полученным от оракулов (подрядчиков) векторам $x_i(y_i)$ технических характеристик системы оценивает доход $\varphi_i(x_i(y_i)) = \varphi_i(y_i)$, на который можно рассчитывать, если выполнение заказа будет поручено i -му исполнителю. Вид функции $\varphi_i(x_i(y_i))$ может быть заранее неизвестен, однако ее значения для заданных значений аргументов могут быть получены экспертным путем. Прибыль, на которую может рассчитывать заказчик при данном варианте предложения $y = (y_1, \dots, y_s)$, равна $g(y) = \max_{1 \leq i \leq s} \{\varphi_i(y_i) - y_i\}$. Задача заказчика — выбрать исполнителей и объем ассигнований, обеспечивающий ему максимальную прибыль. Решение этой задачи сводится к параллельному решению s одномерных задач

$$\varphi_i(y_i) - y_i \rightarrow \max \mid 0 \leq y_i \leq Y$$

и к сравнению оптимальных значений целевых функционалов. Диалоговая процедура решения одномерных задач с оракулами нулевого порядка не представляется трудоемкой.

8. Рассмотрим роль диалогового программирования в корректировке моделей реальных задач.

Пусть начальная модель синтезируемой системы представляет собой задачу P_0 . Может оказаться, что некоторые ограничения исходной содержательной (выпуклой) задачи \bar{P} «по забывчивости» или из-за недостатка на начальном этапе информации об условиях функционирования системы не включены в P_0 (для простоты будем считать, что f_0 априори известна).

Предположим, что на каждом шаге диалога оракулу может быть предъявлена система x , по поводу которой он (оракул) либо сообщает, что она удовлетворяет ограничениям \bar{P} , либо «вспоминает» (и сообщает организатору диалога) новое ограничение задачи \bar{P} , исключающее реализа-

цию системы x . Таким образом, после k шагов диалога задача P_0 уточняется. В новой задаче P_k число ограничений больше, чем в P_0 .

Представляются целесообразными две стратегии организатора диалога.

(1) Стратегия оптимиста — указывать оракулу на $k+1$ -м шаге в качестве очередной системы x_{k+1} , подлежащей анализу, решение задачи P_k (организатор диалога располагает полной информацией о задаче P_k). Оптимист верит, что на данный момент оракул ему уже сообщил все о \bar{P} . Стратегия оптимиста хороша, если задача \bar{P} не очень сложна, т. е. если оракул «забыл» не очень много ограничений. Вместе с тем стратегия оптимиста не гарантирует приближения x к решению задачи \bar{P} .

(2) Стратегия пессимиста — указывать оракулу на $k+1$ -м шаге в качестве очередной системы x_{k+1} центр тяжести (читатель, разобравшийся в механизме МЦТ, поймет смысл выбора центра тяжести области в качестве x_{k+1}) области

$$\{x \in G \mid x \in DP_k, f_0(x) \leq a_k\},$$

где DP_k — множество допустимых планов задачи P_k , а a_k — наилучшее достигнутое к k -му шагу значение целевого функционала f_0 .

Пессимист не желает принимать никаких допущений о том, насколько полно оракул на данном шаге освоил задачу \bar{P} . Он основывает свои решения только на твердо установленных фактах. Стратегия пессимиста гарантирует достижение решения задачи \bar{P} заданной точности за фиксированное время, но не снижает этого времени даже тогда, когда $\bar{P} = P_0$ (и в тех случаях, когда можно с самого начала точно решить задачу \bar{P} , если пользоваться стратегией оптимиста).

По-видимому, рационально синтезировать обе стратегии, например, на четных шагах диалога быть оптимистом, а на нечетных — пессимистом. Время, необходимое для построения решения задачи \bar{P} требуемой точности, не превышает удвоенного минимума таких времен для «чистых» стратегий. Таким образом, синтез стратегий наследует положительные стороны обоих подходов. Однако склонность к компромиссу удваивает трудоемкость процедуры.

Представляется, что применения диалогового программирования будут стимулировать развитие теории информационной сложности.

СЛОЖНОСТЬ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ИНФОРМАЦИИ

Характеристики сложности статистических систем

1. Для изучения закономерностей механизма, определяющего состояние и поведение многих систем в экономике, социальных науках, биологии и технике, исследователь располагает часто лишь результатами наблюдений за реализациями состояний их элементов. Поведение таких систем носит сильно стохастический характер и в деталях слабо предсказуемо. Тем не менее агрегированные характеристики многих явлений и процессов, описываемых в терминах этих систем, оказываются в условиях неизменной или слабо меняющейся среды статистически устойчивыми. Статистическая устойчивость агрегированных характеристик сложных явлений и процессов — основа для прогнозов, без которых невозможно планирование, управление и проектирование.

Один из возможных и достаточно эффективных подходов к анализу систем, заданных статистическими материалами наблюдений и, возможно, некоторой дополнительной априорной информацией, состоит в моделировании их многомерными случайными величинами или процессами. Каждой такой системе приводится в соответствие некоторый случайный вектор, составляющие которого описывают поведение элементов системы или некоторые ее характеристики. Накопленные результаты статистических наблюдений за элементами и/или характеристиками изучаемой системы рассматриваются как реализации случайных значений компонент модели и являются исходным материалом для описания системы и установления закономерностей ее поведения и развития.

Будем называть системы, для которых многомерная случайная величина (или многомерный случайный процесс) является подходящей моделью поведения, статистическими системами. Изучение взаимосвязей между составляющими случайного вектора, анализ различных его ста-

тистических характеристик позволяют установить закономерности состояния и функционирования моделируемой им статистической системы. Это трудоемкий, но необходимый этап в задачах принятия аргументированных решений.

Таким образом, идентификация многих систем по результатам наблюдения за их функционированием часто формально сводится к изучению статистических закономерностей случайного вектора — модели исследуемой системы, компонентами которого являются значения признаков, определяющих систему. Более того, задачи анализа состояния и поведения статистической системы, так же, как и задачи синтеза (в частности, совершенствования и оптимизации) системы, зачастую требуют знания совместной функции распределения вероятностей набора взаимосвязанных случайных величин — наблюдаемых признаков системы или по крайней мере некоторых их статистических характеристик. В задачах количественного анализа вычисляются усредненные значения показателей качества системы и функций, определяющих ограничения ее состояний и поведения. Усреднение этих показателей, вычисление их рассеивания, определение непосредственных связей между ними производятся в соответствии с совместной функцией распределения вероятностей компонент случайного вектора, индуцированного изучаемой статистической системой. Для установления причинно-следственных связей между элементами сложной системы использование многомерных функций распределения вероятностей компонент случайного вектора — модели статистической системы — безусловно, предпочтительней оперирования исходной статистической информацией. Для постановки и решения задач синтеза системы — совершенствования ее структуры или характера функционирования — во многих случаях также необходимо знание совместной функции распределения вероятностей компонент ее модели. Достаточно адекватные постановки задач синтеза сложных систем представляют собой модели стохастического программирования или стохастического оптимального управления, в которых оптимизируется усредненное (в том или ином заранее определенном смысле) значение показателя качества системы при соблюдении (всегда, почти всегда, с некоторой заданной вероятностью или опять-таки в среднем) некоторых заранее предписанных ограничений на возможные состояния системы или поведение ее элементов. Усреднение целевого функционала и, если нужно, области его определения, установ-

ление вероятностных ограничений задачи также производится в соответствии с совместной функцией распределения вероятностей компонент многомерной случайной величины, индуцированной изучаемой статистической системой.

Заметим, что в реальных задачах, с которыми обычно сталкивается исследователь, исходная статистическая информация, как правило, ограничена. Накопленных данных явно недостаточно для того, чтобы без дополнительных знаний или допущений о системе сколь-нибудь надежно восстановить функцию распределения ее компонент.

Мы пришли к выводу, что самые разнообразные задачи изучения и совершенствования статистических систем, информация о функционировании которых накапливается в результате наблюдения за их состояниями и поведением, сводятся к построению и исследованию совместной функции распределения вероятностей компонент многомерной случайной величины, моделирующей набор взаимосвязанных характеристик по ограниченной статистической информации. Различные социально-экономические, технические, биологические и т. п. задачи анализа и синтеза представляют собой характерные задачи изучения и совершенствования систем, закономерности функционирования которых познаются в процессе их статистического исследования. Приведенные соображения свидетельствуют о важности задачи восстановления совместной функции распределения вероятностей компонент случайного вектора, индуцированного изучаемой статистической системой по информационным массивам результатов наблюдений за реализациями ее признаков.

2. В дальнейшем под «анализом статистической системы» мы будем понимать обработку результатов наблюдений за состоянием и поведением системы. Без этого этапа не обходится ни одна проблема принятия решений. Задачей анализа статистической системы является приближенное построение по наблюдениям за ее поведением совместной функции распределения случайного вектора, моделирующего систему. Важно также выявить частные статистические характеристики исходной информации, в терминах которых могут быть эффективно сформулированы, количественно оценены и, что не менее существенно, качественно истолкованы взаимосвязи компонент исследуемой системы. По-видимому, именно поэтому в последнее время значительная часть работ по математическому обеспечению статистической обработки информации и идентификации стохастиче-

ческих явлений и процессов посвящена разработке статистических характеристик связей, непосредственных связей, существенных связей, силы связи и т. д.

Возникает естественный вопрос: какова трудоемкость анализа современных статистических систем — систем достаточно большой размерности — при тех или иных требованиях к точности и достоверности вычисления их характеристик? Другими словами, какое число наблюдений за реализациями состояний элементов системы следует провести, чтобы гарантировать требуемую точность и надежность вычисления статистических характеристик системы? Мы пришли к необходимости оценки сложности статистической обработки результатов наблюдений.

Так же, как и сложность вычислений, сложность статистической обработки информации не может быть оценена единственным показателем. Мы вынуждены ввести ряд характеристик сложности. Трудоемкость статистической обработки результатов наблюдений характеризует сложность оценивания статистических систем, или, что то же самое, сложность описания системы. Мы увидим, что здесь нельзя обойтись одним числом. Сложность оценивания статистической системы зависит от выбора «метода описания системы» и от того, как понимать невязку — «расстояние» между истинной функцией распределения и ее аппроксимацией. Что касается первого фактора — зависимости сложности оценивания от «метода описания системы», — то от его учета можно будет отказаться. Мы приведем некоторые соображения, свидетельствующие о том, что при естественных методах описания системы сложность оценивания существенно не зависит от них. Однако выбор «нормы», по которой оценивается погрешность в аппроксимации функции распределения по результатам наблюдений, радикально влияет на число реализаций, обеспечивающее требуемую точность и надежность оценивания. Поэтому приходится иметь дело с целым спектром характеристик сложности оценивания.

В зависимости от содержательных особенностей и назначения конкретных статистических систем то или иное определение нормы погрешности аппроксимации оказывается более подходящим. Мы увидим, да это и достаточно очевидно, что для получения сколь-нибудь точного и надежного описания произвольной системы немалой размерности необходимо иметь огромное количество реализаций моделирующего ее случайного вектора — гораздо больше, чем это может быть обеспечено при любых предвидимых воз-

возможностях накопления информации. Стало быть, восстановление функции распределения, отвечающей произвольной статистической системе немалой размерности, по результатам наблюдений за ней — нереальная задача. Это значит, что изучение статистических систем не очень малой размерности невозможно, да и не нужно проводить на одном лишь доступном количественному измерению статистическом материале. Исследование закономерностей механизмов, определяющих их состояние, требует привлечения дополнительных априорных соображений об особенностях функционирования систем изучаемого класса. Множество допустимых функций распределения должно быть сужено. Чем больше, однако, принимается априорных качественных предположений, тем выше, естественно, риск появления субъективных оценок и рекомендаций.

В дальнейшем мы введем некоторые ограничения на классы изучаемых статистических систем. В содержательных терминах выделенные статистические системы могут быть представлены композицией подсистем значительно меньшей размерности так, что все (или почти все) существенные связи между элементами системы оказываются внутренними связями подсистем. Будем называть такие статистические системы **блочными**. Представляется, что гипотезе блочности удовлетворяют многие статистические системы большой размерности в экономике и обществе, в биологии и в медицине, в различных отраслях техники. Аппроксимация статистической системы блочной системой **радикально** снижает требования к объему эмпирического материала, необходимого для построения ее модели с заданной точностью и требуемой надежностью. Принятая гипотеза интуитивно приемлема для широкого класса реальных систем. Естественный характер гипотезы оправдывается также следующими соображениями.

Закономерности состояний и поведения многих реальных статистических систем большой размерности установлены на весьма ограниченном эмпирическом материале, явно недостаточном (с формально-статистической точки зрения) для сколь-нибудь серьезных выводов. Тем не менее последующее практическое использование выявленных таким образом закономерностей подтвердило их справедливость. Между тем во многих случаях никакой дополнительной априорной информации о механизме действия системы использовано не было. Статистические закономерности были установлены только по результатам наблюдений. Правдо-

подобное объяснение наблюдаемого феномена заключается в том, что изучаемые статистические системы являлись или по крайней мере хорошо аппроксимировались блочными системами.

Дальше будут формализованы классы статистических систем, удовлетворяющих дополнительным ограничениям. Степень удовлетворения гипотезе блочности будет охарактеризована некоторыми параметрами — существенной размерностью и дефектом информативности системы. Введенные параметры определяют естественную характеристику невязки между истинной функцией распределения и ее аппроксимацией, построенной по результатам наблюдений за статистической системой. Это так называемая мера информационного расхождения Хинчина — Кульбака — Лейбнера. Сложность оценивания, в которой роль нормы — «расстояния» между сравниваемыми функциями распределения — играет информационное расхождение, будем называть **информативной сложностью** статистических систем. Представляется, что для реальных статистических систем это достаточно объективная характеристика трудоемкости статистической обработки информации.

Приведем в заключение несколько примеров статистических систем. В социально-экономических исследованиях часто встречается система «демография — трудовые ресурсы». Элементами этой системы являются показатели демографической ситуации и параметры распределения трудовых ресурсов по отраслям народного хозяйства. Реализациями случайной величины, моделирующей систему, являются данные переписей населения, социологических обследований или статистических учетов в различных регионах страны в разные годы.

Основным объектом медицины являются статистические системы типа «человек — болезнь — метод лечения». Элементами таких систем являются характеристики пациента и параметры, определяющие его состояние здоровья, этапы лечения и течение болезни. Информация о состоянии системы накапливается в результате обследований больных и анализа историй болезни.

Анализ каждой из этих систем на достаточно большом статистическом материале дает основание считать, что их структуры в первом приближении удовлетворяют гипотезе блочности.

1. Как уже отмечалось, отсутствие априорных гипотез, основанных на предварительном качественном анализе, практически исключает возможность установления закономерностей функционирования статистической системы (немалой размерности) на основе данных статистического обследования. Объем наблюдений, необходимый для сколь-нибудь надежной аппроксимации совместного распределения вероятностей компонент случайного вектора — модели системы, астрономически быстро растет с ростом размерности системы. Действительно, пусть размерность системы равна n , а количество градаций (число возможных значений) каждого признака — m . Тогда сетка аргументов, в которых требуется вычислить значения совместной функции распределения вероятностей компонент модели системы, содержит m^n узлов. Чтобы по результатам эксперимента или наблюдений за реальной системой получить статистически достоверную информацию о значениях функции распределения вероятностей взаимосвязанных признаков, определяющих систему, и тем самым идентифицировать изучаемую систему, необходимо в каждом узле сетки аргументов получить по крайней мере несколько измерений. Это значит, что для построения совместной функции распределения вероятностей компонент соответствующего случайного вектора, индуцируемого данной системой, нужно располагать числом измерений порядка $O(m^n)$. При $m=5$ и $n=20$ или при $m=2$ и $n=50$ практически невозможно реализовать m^n измерений в приемлемые сроки при любом уровне развития техники измерения. Следует при этом учесть, что значения m и n в реальных задачах планирования и управления социально-экономическим развитием, проектирования технических систем и т. п. могут значительно превосходить указанные величины.

Априорные гипотезы, выделяющие класс статистических систем, могут отражать как содержательные, качественные свойства систем, входящих в этот класс, так и формальные особенности индуцируемых ими многомерных случайных величин. Так, например, класс статистических систем может определяться структурой графа связей между элементами систем класса или какими-нибудь другими характеристиками взаимодействия элементов системы. В формальных терминах класс статистических систем можно задать функциональным видом совместных распределений вероят-

ностей компонент случайных векторов, моделирующих системы класса, заданным с точностью до численных значений параметров распределений. Например, можно рассматривать класс статистических систем, моделируемых нормально распределенными многомерными случайными величинами.

2. В реальных задачах исследования статистических систем наиболее трудоемким и дорогостоящим этапом является сбор статистической информации. Поэтому желательно заранее представлять себе необходимый объем статистической информации для идентификации системы. В связи с этим одной из важных задач теории статистических систем является оценка сложности их описания.

Естественно под **сложностью оценивания (описания)** $N_A(n, \varepsilon, \alpha)$ класса n -мерных статистических систем понимать минимальное число независимых, одинаково распределенных (в соответствии с истинным распределением вероятностей) реализаций моделирующей систему n -мерной случайной величины, необходимое для того, чтобы с заданной надежностью α восстановить совместную функцию распределения вероятностей компонент модели любой статистической системы из класса A с погрешностью, не превосходящей заданную величину ε .

Для формального определения сложности оценивания класса статистических систем A целесообразно вначале ввести понятие «метод B описания многомерной случайной величины». Дело в том, что сложность оценивания зависит от того, как используются результаты наблюдений за исследуемой статистической системой для формирования совместного распределения вероятностей компонент ее модели. Естественно связывать сложность оценивания с самым экономным методом описания.

Определим, следуя [51], **метод B описания** многомерной случайной величины $\xi \in A$ по N независимым одинаково распределенным (в соответствии с истинным распределением вероятностей $p(x)$ случайного вектора ξ) наблюдениям $x^N = (x_1, \dots, x_N)$, как **любое** отображение $B: X^N \rightarrow A$, где X — пространство элементарных событий (множество, на котором определены функции распределения вероятностей $p(x)$ случайных величин $\xi \in A$, $X^N = \underbrace{X \times \dots \times X}_{N \text{ раз}}$. Выражение

«метод B построения функции распределения вероятностей (описания статистической системы) имеет $\|\cdot\|$ -погрешность ε

при надежности α на классе случайных векторов (статистических систем) A означает, что

$$P \{x^N | \|p_B(x | x^N) - p(x)\| \leq \varepsilon\} \geq \alpha \quad \forall \xi \in A.$$

Другими словами, метод B имеет $\|\cdot\|$ -погрешность, не превосходящую ε с надежностью α , если вероятность того, что в результате работы метода B с выборкой размера N получится распределение вероятностей, отличающееся от истинного по мере $\|\cdot\|$ не более чем на ε , не меньше α . Метод B , имеющий на классе случайных векторов A при N наблюдениях $\|\cdot\|$ -погрешность не более ε при надежности не менее α , будем обозначать через $B_N(\|\cdot\|, \varepsilon, \alpha)$.

3. Теперь мы можем дать строгое формальное определение сложности оценивания класса A многомерных случайных величин (класса статистических систем, моделируемых этими случайными векторами). Будем различать сложность оценивания, отвечающую заданному методу B описания класса A статистических систем, и сложность оценивания, определяемую только классом рассматриваемых систем.

Под сложностью оценивания (описания) $N_A(B, \|\cdot\|, \varepsilon, \alpha)$ с заданными $\|\cdot\|$ -погрешностью ε и надежностью α , отвечающей фиксированному методу B описания статистических систем класса A , будем понимать минимальное число N независимых, одинаково распределенных (в соответствии с истинной функцией распределения вероятностей) наблюдений, гарантирующих восстановление по методу $B \equiv p_B(x | x^N)$ совместной функции распределения вероятностей $p(x)$ компонент случайного вектора $\xi \in A$, принимающего значения $x \in X$ с $\|\cdot\|$ -погрешностью не более ε и надежностью не менее α , т. е.

$$\begin{aligned} N_A(B, \|\cdot\|, \varepsilon, \alpha) &= \\ &= \min \{ N | P [\|p_B(x | x^N) - p(x)\| \leq \varepsilon] \geq \alpha, \forall \xi \in A \} = \\ &= \min \{ N | B = B_N(\|\cdot\|, \varepsilon, \alpha) \}. \end{aligned}$$

Сложность оценивания (описания) $N_A(\|\cdot\|, \varepsilon, \alpha)$ класса A n -мерных статистических систем (сложность идентификации совместного распределения вероятностей n -мерной случайной величины $\xi \in A$) при $\|\cdot\|$ -погрешности ε и надежности α) определим как минимальное число N независимых, одинаково распределенных (в соответствии с истинной функцией распределения вероятностей) реализаций наблюдений за исследуемой статистической системой, при котором еще существует метод описания $B = p_B(x | x^N)$, позволяющий с

$\|\cdot\|$ -погрешностью, не превышающей ε , и надежностью не менее α восстановить истинное распределение вероятностей $p(x)$ для любого случайного вектора $\xi \in A$, т. е.

$$N_A(\|\cdot\|, \varepsilon, \alpha) = \min \{ N \mid \exists B: B = B_N(\|\cdot\|, \varepsilon, \alpha) = \\ = \min_B N_A(B, \|\cdot\|, \varepsilon, \alpha). \}$$

Сложность оценивания класса статистических систем (случайных векторов, моделирующих системы класса) зависит не только от числа элементов системы (ее размерности), но и от количества возможных состояний этих элементов (количества градаций компонент случайного вектора, индуцированного системой). Чтобы не увеличивать число параметров, определяющих сложность оценивания класса A статистических систем, целесообразно устанавливать зависимость сложности оценивания N_A от числа $M = |X|$ различных векторов $x \in X$ возможных значений, принимаемых случайной величиной $\xi \in A$ ($M = \prod_{i=1}^n m_i$, где m_i — количество градаций i -й компоненты случайного вектора ξ).

4. Будем рассматривать класс A статистических систем, моделями которых служат дискретные многомерные случайные величины ξ с совместным распределением вероятностей $p(x)$, определенным на конечном множестве X из M элементов. Заметим, что множество всевозможных распределений вероятностей, определенных на X , представляет собой симплекс в R^M .

Поскольку сложность описания $N_A(\|\cdot\|, \varepsilon, \alpha)$ класса A статистических систем зависит от выбора меры близости между распределениями, то представляет интерес вопрос о том, как она изменяется с изменением нормы на R^M . Пусть $\|\cdot\|$ и $\|\cdot\|'$ две нормы на R^M . Положим, по определению $\beta = \max_{\xi \in R^M} \frac{\|\xi\|}{\|\xi\|'}$. Тогда справедливо следующее неравенство [51]:

$$N_A(\|\cdot\|, \varepsilon, \alpha) \leq N_A(\|\cdot\|', \frac{\varepsilon}{\beta}, \alpha).$$

Для лебеговых метрик $\|\cdot\| = \left[\sum_{i=1}^M |\xi_i|^p \right]^{\frac{1}{p}}$ (частным случаем которых при $p=2$ является обычная эвклидова мет-

рика) можно явно определить значение β :

$$\beta = \max_{\xi \in R^M} \frac{\|\xi\|_q}{\|\xi\|_p} = \begin{cases} M^{\frac{1}{q} - \frac{1}{p}}, & p > q, \\ 1, & p \leq q. \end{cases}$$

Установленное неравенство позволяет проводить оценку сложности описания статистических систем в какой-либо одной фиксированной метрике, например в метрике $\|\xi\|_\infty = \max_{1 \leq i \leq n} |\xi_i|$. Для любой другой метрики оценку можно получить из соотношения

$$N_A(\|\cdot\|, \varepsilon, \alpha) \leq N_A\left(\|\cdot\|_\infty, \frac{\varepsilon}{\beta}, \alpha\right),$$

при этом если $\|\cdot\| = \|\cdot\|_p$, то $\beta = M^{1/p}$.

5. Для того чтобы получить оценку сверху сложности идентификации распределений вероятностей случайных векторов, моделирующих статистические системы класса A , достаточно определить сложность идентификации систем класса по какому-нибудь конкретному методу описания B статистических систем.

Наиболее распространенным методом описания статистических систем является метод аппроксимации распределения вероятностей компонент модели системы частотой $B = \rho_B(x | x^N) = \frac{N_x}{N}$, где N_x — число наблюдений, при которых значения компонент случайной величины ξ (соответственно значения элементов системы, моделируемой ξ) определяются вектором $x \in X$. Будем называть этот метод описания статистических систем **методом частот**.

Другой метод (точнее, класс методов) описания статистических систем состоит в следующем. Наблюдения, отвечающие одним и тем же условиям, разбиваются по какому-то принципу на группы. В каждой группе оценки определяются методом частот. В качестве оценки вероятностей $p(x)$ значения случайной величины в точке x принимается медиана оценок по группам наблюдений. Методы этого класса будем называть **методами медианы**.

В [51] получены приближенные выражения для сложности оценивания статистических систем, отвечающие разным методам описания систем и разным мерам невязки между истинным распределением и восстановленным по результатам наблюдений. Имеют место следующие утверждения.

(1). Сложность оценивания по методу частот, обеспечивающая $\|\cdot\|_\infty$ -погрешность не более ε с надежностью не менее α , определяется величиной

$$N^{(\infty)} \approx \frac{2}{\varepsilon^2} \max_{x \in X} \{p(x)(1-p(x))\} \ln \frac{1}{1-\alpha}$$

и при любых $p(x)$ не превышает

$$N^{(\infty)} = \frac{1}{2\varepsilon^2} \ln \frac{1}{1-\alpha}.$$

Отсюда для класса $A_\sigma (\sigma \leq \frac{1}{2})$ статистических систем

$$A_\sigma = \{p(x) \mid p(x)(1-p(x)) \leq \sigma^2\}$$

имеем

$$N_{A_\sigma}^{(\infty)} \leq \frac{2\sigma^2}{\varepsilon^2} \ln \frac{1}{1-\alpha}.$$

По сложности оценивания в метрике $\|\cdot\|_\infty$ получаем в соответствии с приведенными выше неравенствами верхние оценки сложности оценивания по методу частот для любой лебеговой метрики $\|\cdot\|_p$. Имеем

$$N^{(p)} \leq \frac{2M^{2/p}}{\varepsilon^2} \max_{x \in X} \{p(x)(1-p(x))\} \ln \frac{1}{1-\alpha} \leq \frac{M^{2/p}}{2\varepsilon^2} \ln \frac{1}{1-\alpha}.$$

$$N_{A_\sigma}^{(p)} \leq \frac{2M^{2/p}}{\varepsilon^2} \sigma^2 \ln \frac{1}{1-\alpha}.$$

(2). Сложность оценивания по методам медианы, обеспечивающая $\|\cdot\|_\infty$ -погрешность не более ε с надежностью не менее α , определяется величиной

$$N^{(\infty)} \approx \frac{2e}{\varepsilon^2} \max_{x \in X} \{p(x)(1-p(x))\} \ln \frac{1}{2(1-\alpha)} \leq \frac{e}{2\varepsilon^2} \ln \frac{1}{2(1-\alpha)},$$

$$N_{A_\sigma}^{(\infty)} \leq \frac{2e}{\varepsilon^2} \sigma^2 \ln \frac{1}{2(1-\alpha)}.$$

Для сложности оценивания в метрике $\|\cdot\|_p$ по методам медианы справедливы следующие неравенства:

$$N^{(p)} \leq \frac{2eM^{2/p}}{\varepsilon^2} \max_{x \in X} \{p(x)(1-p(x))\} \ln \frac{1}{2(1-\alpha)} \leq \frac{eM^{2/p}}{2\varepsilon^2} \ln \frac{1}{2(1-\alpha)},$$

$$N_{A_\sigma}^{(p)} \leq \frac{2e}{\varepsilon^2} \sigma^2 M^{2/p} \ln \frac{1}{2(1-\alpha)}.$$

Полученные оценки могут быть использованы, например, для вычисления количества наблюдений, позволяющих проверить с $\|\cdot\|_p$ -погрешностью не более ε и надежностью не менее α гипотезу о том, что распределение изучаемой случайной величины имеет заданный вид. В частности, для проверки гипотезы о равномерном распределении случайной величины x при оценке невязки между истинным и эмпирическим распределениями в эвклидовой метрике ($p=2$) достаточно ограничиться числом наблюдений, не превышающим

$$N^{(2)} \approx \frac{2eM}{\varepsilon^2} \left\{ \frac{1}{M} \left(1 - \frac{1}{M} \right) \right\} \ln \frac{1}{2(1-\alpha)} \approx \frac{2e}{\varepsilon^2} \ln \frac{1}{2(1-\alpha)}.$$

Сравнивая установленные верхние грани для сложности оценивания при разных методах описания статистических систем, видим, что они величины одного порядка. Это дает основание предполагать, что сложность оценивания будет слабо зависеть от метода описания статистической системы и будет указанным образом зависеть от M , ε и α .

Информативная сложность

1. Определение сложности оценивания, связанное с нормой отклонения приближенного распределения от истинного, слабо приспособлено к учету априорной информации о структуре связей в изучаемом классе систем. Гораздо естественней и проще учитывать часто встречающиеся особенности классов статистических систем, если вместо $\|\cdot\|$ -погрешности рассматривать так называемую невязку Хинчина — Кульбака — Лейбера — меру информационного расхождения истинного и приближенного распределений [26]. При этом удастся конструктивно учитывать наиболее естественную априорную информацию об изучаемых классах статистических систем — возможность приближенного представления совместного распределения вероятностей многомерной случайной величины — модели системы — в виде композиции частных распределений небольшой размерности. Это информация о существенных связях системы. Информационное расхождение как мера невязки распределений приспособлено к описанию и оценке трудоемкости анализа статистических систем, которые могут быть аппроксимированы блочными системами.

Известно, что два распределения вероятностей с нулевым информационным расхождением совпадают с точностью до множества меры нуль, т. е. в случае дискретных распределений (которые мы только и рассматривали здесь) просто совпадают. Это еще один аргумент в пользу выбора невязки по Хинчину — Кульбаку — Лейберу в качестве меры близости истинного и приближенного распределений.

Напомним, что для дискретных распределений информационное расхождение распределений вероятностей p и \tilde{p} , определенных на множестве X , вычисляется по формуле

$$I(p: \tilde{p}) = \sum_{x \in X} p(x) \ln \frac{p(x)}{\tilde{p}(x)}.$$

В задачах, в которых следует стремиться к минимальным потерям информации об изучаемой системе, получаемой по результатам наблюдений, информационное расхождение является, по-видимому, наиболее целесообразной оценкой качества аппроксимации статистической системы.

Назовем **информативной сложностью** класса A статистических систем (или соответственно класса A случайных величин) минимальное число K независимых одинаково распределенных (в соответствии с истинной функцией распределения) наблюдений x_1, x_2, \dots, x_K , при которых еще существует метод $B(\varepsilon, \alpha)$, позволяющий по ним построить приближенное распределение вероятностей $p(x|x_1, \dots, x_K) = \tilde{p}(x|x^K)$, аппроксимирующее истинное распределение $p(x)$ с информационной невязкой, не превышающей ε , и с надежностью не менее α .

Формально

$$K_A = \min \{ K \mid \exists B(\varepsilon, \alpha): P[I(p(x): \tilde{p}(x|x^K)) \leq \varepsilon] \geq \alpha \}.$$

Здесь $x^K = (x_1, \dots, x_K)$ — независимые, одинаково распределенные по $p(x)$ наблюдения, а $P(S)$ — вероятность события S .

Таким образом, информативная сложность классов статистических систем — это их сложность оценивания, в которой $\|\cdot\|$ -погрешность аппроксимации заменена информационным расхождением $I(p: \tilde{p})$.

Пусть по-прежнему X — конечное множество из M элементов, A — семейство распределений вероятностей на X . Предполагается, что для всех $p \in A$ выполняется $p(x) > 0 \forall x \in X$. Будем интерпретировать M как число различных

наборов значений n -мерной случайной величины, i -я компонента которой принимает одно из m_i значений. Таким образом, $M = \prod_{i=1}^n m_i$.

В [52] построена оценка сверху для информативной сложности произвольного класса A статистических систем.

Имеет место утверждение, при не слишком малых α ($1 - e^{-\frac{1}{\alpha}} \leq \alpha \leq 1$) и $\epsilon > 0$ справедлива следующая оценка информативной сложности класса статистических систем:

$$K_A(\epsilon, \alpha) \leq \frac{2M}{v^2} \ln \frac{1}{1-\alpha},$$

где v — положительный корень уравнения

$$v^2 = 2\epsilon(1-v).$$

Для оценки сверху информативной сложности K_A использовано значение K для метода частот. Оценки такого же порядка получены и при некоторых других естественных методах описания статистических систем. Это дает основание полагать, что оценку информативной сложности вряд ли можно существенно улучшить.

Обратим здесь внимание читателя на то, что при малых ϵ верхняя оценка информативной сложности растет с увеличением требуемой точности $\frac{1}{\epsilon}$ гораздо медленнее, чем верхняя оценка сложности описания статистической системы: $K = O\left(\frac{1}{\epsilon}\right)$, $N = O\left(\frac{1}{\epsilon^2}\right)$.

2. До сих пор мы не принимали никаких допущений о свойствах изучаемых статистических систем. Величина $M = \prod_{i=1}^n m_i$, входящая множителем в оценку информативной сложности, для реальных статистических систем — астрономическая величина. Это значит, что при отсутствии априорной информации о классе изучаемых систем нельзя ожидать сколь-нибудь качественного построения соответствующих им распределений при любом обозримом наборе наблюдений за реализациями состояний элементов системы. В начале главы была высказана гипотеза о блочной структуре многих реальных систем. Информативная сложность таких систем существенно ниже. Функции распределения,

отвечающие блочным системам достаточно высокой размерности, могут быть практически восстановлены по результатам измерений.

Приведем формальное определение блочных систем и оценим информативную сложность таких систем в зависимости от характеризующих их параметров — существенной размерности и дефекта информативности.

Будем называть n -мерную статистическую систему системой k -существенной размерности, если совместная функция распределения модели системы может быть представлена в виде композиции ее частных распределений размерности не выше k . Точнее, n -мерная статистическая система является системой k -существенной размерности, если существует перестановка $\alpha = \{\alpha_i\}_{i=1}^n$ индексов $\{1, 2, \dots, n\}$ ее элементов и наборы $\Delta(\alpha_i)$ индексов, удовлетворяющие условию $\Delta(\alpha_i) \subset \{\alpha_1, \alpha_2, \dots, \alpha_{i-1}\}$ и содержащие не более чем по $k-1$ индексов ($|\Delta(\alpha_i)| \leq k-1$), таких, что совместное распределение $p(x_1, \dots, x_n)$ случайных величин, моделирующих элементы системы, может быть представлено в виде произведения

$$p(x) = \prod_{i=1}^n p(x_{\alpha_i} | x_{\Delta(\alpha_i)}) = p(x_{\alpha_1}, \dots, x_{\alpha_k}) \prod_{i=k+1}^n p(x_{\alpha_i} | x_{\Delta(\alpha_i)}).$$

Таким образом, существенная размерность статистической системы — число k определяется по формуле

$$k = \min_{\{\alpha_i\}_{i=1}^n} \max_{1 \leq i \leq n} |\Delta(\alpha_i)| + 1,$$

где $\{\alpha_i\}_{i=1}^n$ — перестановка индексов $\{1, 2, \dots, n\}$; $\Delta(\alpha_i)$ — набор индексов из $\{\alpha_1, \dots, \alpha_{i-1}\}$; $|\Delta(\alpha_i)|$ — число индексов в $\Delta(\alpha_i)$.

Каждая статистическая система характеризуется двумя размерностями: одной явной, равной числу n элементов системы, и другой скрытой — существенной, определяемой наиболее экономным блочным представлением системы.

Будем обозначать множество всех распределений вероятностей n -мерных случайных векторов, определенных на X , через P . Подмножество распределений, представимых в виде композиций частных распределений, размерности не выше k , обозначим через P_k . Распределения из P_k моделируют статистические системы k -существенной размерности. Статистические системы, отвечающие распределениям из P_k , будем называть k -блочными.

Понятие k -блочных систем дает основание для классификации статистических систем и выделения* классов систем, поддающихся статистическому анализу при обозримом объеме наблюдений за состояниями ее элементов. Качество аппроксимации реальной системы k -блочной системой естественно определять посредством информативной невязки $I(p : \tilde{p}_k)$ между соответствующими распределениями вероятностей.

В [49] приведена схема построения (по результатам наблюдений) наилучшей информативной аппроксимации заданной n -мерной статистической системы системой k -существенной размерности. Схема вычислений сводится к решению специальной задачи линейного целочисленного программирования. Для $k=2$ имеется эффективный метод решения этой задачи. При $k>2$ задача оказывается NP -полной. Однако в [50] для нее построен достаточно простой асимптотически оптимальный метод решения.

Обозначим

$$I(p : P_k) = \min_{\tilde{p} \in P_k} I(p : \tilde{p}).$$

Статистическая система, моделируемая n -мерным совместным распределением вероятностей $p(x)$, является системой k -существенной размерности в том и только в том случае, если $I(p : P_k) = 0$.

Статистические системы размерности n , для которых минимальная информационная невязка между ее моделью — распределением p и распределением $\tilde{p} \in P_k$ не превосходит некоторого числа β_k , т. е. $I(p : P_k) \leq \beta_k$, будем называть системами k -существенной размерности с дефектом информативности β_k , или, короче, системами с дефектом информативности β_k .

При фиксированном дефекте информативности β существенная размерность k_β , характеризующая статистическую систему, определяется по формуле

$$k_\beta = \min \{k \mid I(p : P_k) \leq \beta\}.$$

Естественно, что конструктивные методы анализа можно ожидать лишь для тех статистических систем, у которых $k_\beta \ll n$, т. е. для систем малой существенной размерности.

3. Приведем оценку информативной сложности систем k -существенной размерности и систем, аппроксимируемых ими.

Пусть P — множество всех n -мерных распределений вероятностей, определенных на $X = (x_1, \dots, x_M)$, $P_k \subset P$ — подмножество распределений из P , которые могут быть представлены композицией распределений размерности не выше k , т. е. класс распределений k -существенной размерности; $P_{k\beta}$ — класс распределений k -существенной размерности с дефектом информативности β . $P_{k\beta} = \{p \in P \mid I(p; P_k) \leq \beta\}$. $P_{k\beta}$ — это класс систем, которые могут быть с малым информационным расхождением β аппроксимированы статистическими системами из P_k .

Ясно, что $P_k = P_{k0}$.

Введем величину M_k

$$M_k = \max \left\{ \prod_{i \in \{\alpha\}} m_i \mid |\{\alpha\}| \leq k \right\}.$$

Здесь $\{\alpha\} \subset \{1, \dots, n\}$; $|\{\alpha\}|$ — число элементов в $\{\alpha\}$; m_i — число градаций i -й составляющей случайного вектора, моделирующего статистическую систему.

В [52] построена оценка сверху для информативной сложности класса $P_{k\beta}$ статистических систем k -существенной размерности с дефектом информативности β . Имеет место утверждение. При не слишком малых α ($1 - e^{-\frac{1}{\alpha}} \leq \alpha \leq 1$) и $\epsilon > \beta \geq 0$ справедлива следующая оценка:

$$K_{P_{k\beta}}(\epsilon, \alpha) \leq \frac{2M'_k}{v_k^3} \ln \frac{1}{1-\alpha},$$

где v_k — положительный корень уравнения

$$\frac{v_k^3}{2(1-v_k)} = \frac{\epsilon - \beta}{n - k + 1}.$$

Таким образом, дополнительная информация о том, что n -мерная статистическая система может быть аппроксимирована с дефектом информативности β системой k -существенной размерности, позволяет при $k \ll n$ радикально сократить объем эмпирического материала, необходимого для построения модели системы (совместного распределения вероятностей состояний ее элементов) с заданной точностью (информативной невязкой) и требуемой надежностью. Например, для построения совместного распределения вероятностей составляющих 100-мерного дихотомического случайного вектора (т. е. 100-мерного случайного вектора, каждая компонента которого может принимать лишь два

значения) с информативной невязкой $\varepsilon=0,25$ и надежностью $\alpha=0,9$ требуется порядка 10^{31} наблюдений за системой, моделируемой этим вектором. Ясно, что такое количество реализаций многомерной случайной величины получить практически невозможно. В то же время, если имеется дополнительная информация, позволяющая отнести систему к классу $P_{k\beta}$, например, с $k=3$ и $\beta=0,05$, то при тех же ε и α для построения информативной аппроксимации распределения 100-мерного дихотомического вектора требуется наблюдать менее 10^4 его реализаций, а при $k=10$ и $\beta=0,05$ — не более 10^6 реализаций.

Интуитивно представляется правдоподобным, что многие реальные системы высокой размерности приближаются при небольшом дефекте информативности системами с малой существенной размерностью.

Выводы

Сложность обработки результатов наблюдений очень быстро растет с ростом размерности соответствующей статистической системы. Каждая статистическая система характеризуется двумя размерностями. Первая задается явно. Это характеристика класса систем — число переменных в модели системы. Вторая — скрытая. Она зависит от структуры конкретной системы и определяется так называемой существенной размерностью системы.

Статистическая система, точнее, ее модель — совместное распределение вероятностей $p(x)$ состояний ее элементов — может быть описана многими различными способами:

$$p(x) = \prod_{i=1}^n p(x_{\alpha_i} | x_{\Delta(\alpha_i)}),$$

где $\{\alpha_i\}_{i=1}^n$ — некоторая перестановка набора $\{1, 2, \dots, n\}$ индексов элементов системы, а $\Delta(\alpha_i) \subset \{\alpha_1, \dots, \alpha_{i-1}\}$. Среди этих записей имеется одна (иногда несколько), которая определяет существенную размерность системы. Она соответствует перестановке, на которой достигается $\min \max_{1 \leq i \leq n} |\Delta(\alpha_i)|$. Эта перестановка определяет пред-

ставление $p(x)$, отвечающее структуре конкретной статистической системы и выявляющее ее существенные связи.

Информативная сложность системы, вычисленная по ее

явной размерности n — по числу элементов системы, определяет трудоемкость статистической обработки класса P всех n -мерных систем, т. е. число наблюдений, позволяющее получить с заданной точностью и требуемой надежностью функции распределения любой статистической системы из n элементов. Информативная сложность, вычисленная по ее существенной размерности k , характеризует трудоемкость статистической обработки систем (подкласса P_k), обладающих k -блочной структурой. Как правило, она значительно ниже информативной сложности класса P .

Многие реальные статистические системы могут быть достаточно хорошо аппроксимированы (с малой потерей информации — с некоторым дефектом информативности) системами весьма малой существенной размерности. Информативная сложность любой такой системы, как правило, невелика. Аппроксимация закономерностей механизмов, определяющих поведение соответствующей системы, может быть практически осуществлена по результатам наблюдений за реализациями состояний ее приближенной модели — случайной величины относительно невысокой размерности. Это, пожалуй, один из основных результатов теории информативной сложности.

КАТЕГОРИЯ «СЛОЖНОСТЬ» И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ Проблема «человек — машина»

1. Одна из важных задач теории сложности — прогнозирование возможного будущего, к которому могут привести те или иные решения. В связи с этим разговор о категории «сложность» не может пройти мимо набивших оскомину вопросов что может и чего не может ЭВМ, в какой мере реален «искусственный разум», что можно ожидать от искусственного интеллекта в обозримом будущем и в какой мере следует возлагать надежды на искусственный разум? Короче говоря, завершая наш краткий обзор современного состояния теории сложности, нельзя уклониться от обсуждения проблемы века — «человек и машина»

Вопрос о перспективах передачи интеллектуальных функций человека вычислительной машине является предметом ожесточенных дискуссий между учеными различных

направлений и вызывает неизменный интерес у широкой публики. Мы с самого начала исходим из того, что проблема «машина и мышление» является не только и не столько технической или математической, сколько социологической и этической. По-видимому, обществу важен не только ответ на вопрос, какие человеческие функции **можно передать машине**, но и в гораздо большей степени, какие человеческие функции **нужно ей передать**. Если, например, потенциальные временные ресурсы, необходимые для искусственной реализации функций естественного интеллекта, определяющих, скажем, его «мудрость» и «гуманность», соизмеримы с временем, в течение которого содержание этих понятий в обществе существенно не меняется, то вопрос, сможет ли машина выполнять роль врача или юриста, управляющего производством или начальника штаба соединения, нужно заменить вопросом, следует ли машине доверить соответствующие функции.

Конечно, можно возразить, что обязанности «недостаточно мудрой и гуманной» машины могут быть ограничены консультацией специалистов, анализирующих подготовленные машиной решения. И действительно, трудно переоценить пользу, которую может получить врач или юрист, директор или командующий от машины с богатым банком знаний и высокой производительностью. Не следует, однако, упускать из виду, что темпы роста исследований и использования вычислительной техники в различных областях человеческой деятельности и «головокружение от успехов» ЭВМ могут вывести ее из-под контроля. Приведенные далее высказывания специалистов, имеющих непосредственное отношение к развитию и применению машин, свидетельствуют о том, что это отнюдь не беспочвенные опасения. Социальные последствия бесконтрольного развития искусственного интеллекта не менее опасны, чем последствия неконтролируемых работ по расщеплению атомного ядра или по биологическим экспериментам с искусственными рекомбинантными молекулами ДНК.

2 Вряд ли можно сейчас возражать против того, что любая четко определенная формализуемая функция человеческого разума может быть «в конце концов» алгоритмизована (сведена к вычислениям) и передана вычислительной машине. Во всех таких видах деятельности машина все более успешно конкурирует с человеком. Больше того, огромная производительность и надежная память ставят машину в таких ситуациях часто в более благоприятное положение

ние по сравнению с человеком. Имеется чрезвычайно широкий спектр задач, отвечающих насущным требованиям человечества, успешное решение которых может обеспечить незаменимый усилитель интеллекта — вычислительная машина, и только она. Уже сейчас ЭВМ обеспечивают решение таких проблем, как управление полетами космических ракет, автоматизированное проектирование и производство, разработка технологии создания материалов с заданными свойствами, прогноз погоды. А на очереди — управляемый термоядерный синтез, освоение ресурсов Мирового океана, защита окружающей среды. Все это, однако, задачи, которые либо уже **четко формализованы**, либо, по мнению достойных доверия специалистов, **могут быть в обозримом будущем четко поставлены** и хотя бы «в принципе» формализованы.

Возникает естественный вопрос: все ли понятия, которыми оперируют люди в процессе принятия решений, могут быть четко определены и переведены на язык чисел, либо представлены на каком-либо формальном языке. Можно ли формально определить такие многогранные и многозначные категории, как мораль, справедливость, счастье, культура и другие понятия, выделяющие *homo sapiens* из животного мира, определяющие его личные цели и общественные идеалы. Эти понятия присущи человеку как члену общества и формируются и эволюционируют в процессе исторического развития природы и общества.

3. Приведем два примера, ставящие под сомнение возможности искусственного интеллекта даже не в столь сложных ситуациях выбора решений, которые непосредственно связаны с такими неформализуемыми понятиями, как мораль, этика и т. д.

Первый пример заимствован из книги «Творец и робот», в которой ее автор Норберт Винер, подчеркивая опасность магии автоматизации, приводит новеллу В. В. Джекобса «Обезьянья лапа». Мы используем эту новеллу для несколько иной цели.

Фабула новеллы такова. В английской рабочей семье слушают рассказы сержанта, возвратившегося из Индии. Гость показывает талисман — высушенную обезьянью лапу и говорит, что талисман наделен магическим свойством исполнять три желания своего владельца. Однако талисман, утверждает он, не принес своим владельцам счастья. Наоборот, все бывшие владельцы талисмана, и сержант в их числе, натерпелись бед. Гость уже намерен бросить волшеб-

ную лапу в камин, но хозяин, невзирая на предостережение сержанта, овладевает талисманом и просит обезьянью лапу выполнить три его желания. Первое — получить двести фунтов стерлингов.

Через некоторое время раздается стук в дверь. В комнату входит служащий фирмы и сообщает, что сын хозяина — рабочий фабрики, принадлежащей фирме, погиб в результате несчастного случая. Фирма выражает семье погибшего соболезнование и просит принять пособие в размере двухсот фунтов.

Так выполняется первое желание хозяина. В таком же духе магический талисман выполняет и два других желания. Талисман — своего рода искусственный интеллект — выполняет поставленную перед ним задачу как бездумный исполнитель. Он не уточняет постановку задачи и не дополняет исходную информацию ограничениями, исключающими нежелательные варианты решений. Чтобы подойти «сознательно» к решению задачи, талисман должен был бы знать, что такое «нежелательные решения», в каких условиях допустима реализация решения, какие препятствия могут при этом встретиться. Короче говоря, он должен располагать информацией, которую люди, занимающиеся той же задачей, молчаливо предполагают само собой разумеющейся. Другими словами, он должен был бы обладать некоторым жизненным опытом, а поскольку класс заданий талисману заранее жестко не ограничен, то дополнительная информация, позволяющая ему «разумно» выполнять желания владельца, должна соответствовать опыту вдумчивого исполнителя. Приобрести такой опыт — значит, повторить эволюцию homo sapiens и весь путь его общественного развития. Другого пути мы не видим.

Второй пример был рассказан одному из авторов его старшим товарищем — очевидцем и участником описываемых ниже событий. Дело было в 1916 г. на одном из участков застabilизировавшегося к тому времени Западного фронта. В течение нескольких месяцев активных действий на участке не наблюдалось. Только редкие перестрелки нарушали спокойное течение «окопной войны». Как-то командиру части сообщили, что на его участок собирается с инспекцией великий князь. Было приказано продемонстрировать боеготовность полка и доблесть войск: атаковать противника и занять некоторые его рубежи. Поставленная задача не соответствовала, однако, боезапасу, которым располагала часть. Возникла проблема выбора объектов, по которым

следовало сосредоточить артогонь, и объекта для первого удара, чтобы при ограниченном числе снарядов возможно больше ослабить ожидаемое сопротивление противника. Офицеры штаба высказывали разные предложения. Все они, однако, были слабо аргументированы. Споры затянулись бы, если бы денщик командира полка не попросил разрешения вмешаться в разговор. Наблюдая время от времени в перископ за позициями противника, он заметил, что, как только выглядывает солнце, на крыше одной из землянок появляется и греется на солнце котенок с бантиком. С выводом денщика нельзя было не согласиться: землянка явно была офицерским жильем или командным пунктом. И в том и в другом случае первой целью артналета следовало выбрать землянку с котенком.

Рассказывая эту историю, очевидец событий спрашивал у автора, в то время еще молодого сторонника модных в те годы кибернетических идей, представляет ли он себе автомат (в нынешних терминах искусственный интеллект), способный найти основания для выбора решения, подобно тому как это сделал неграмотный денщик в описанной ситуации. Авторы готовы задать тот же вопрос читателю.

Мы уже указывали, что единственный, с нашей точки зрения, путь накопления опыта принятия решений в житейских ситуациях — это моделирование эволюции и социального развития человека. Правдоподобные соображения о сложности таких процессов излагаются далее.

Объективные и субъективные факторы в принятии решений

В теории принятия решений формализованы и исследованы главным образом те ситуации выбора, в которых известна, по крайней мере на содержательном уровне, цель решения — критерий качества выбора. Это ситуации, описываемые моделями безусловной и условной оптимизации. В предыдущих главах обсуждались характеристики сложности различных классов задач, соответствующих этим моделям. Приведенные там оценки сложности определяют пути и трудоемкость автоматизации решения задач оптимизации при известных критериях качества выбора.

Значительно более серьезные проблемы возникают при постановке и анализе естественных задач принятия решений, в которых требуется обеспечить достижение многих целей,

удовлетворить интересы многих лиц, выбрать взаимоприемлемые стратегии поведения конфликтующих сторон. Речь идет о современных разделах теории принятия решений — теории многокритериальных задач, теории групповых решений и теории игр. Формальная постановка задач этих разделов требует четкого определения понятий «справедливое агрегирование целей», «рациональное согласование интересов», «равновесное разрешение конфликта». Существует множество, казалось бы, удовлетворительных определений этих понятий, приводящих, вообще говоря, к различным и иногда существенно различным решениям. Выбор подходящего определения зависит обычно от огромного количества содержательных признаков ситуации. Перечень признаков (и характер зависимости от них) интуитивных принципов компромисса, как правило, не удастся заранее учесть и классифицировать. Они существенно определяются конкретными особенностями задачи. В общем случае нет формальных оснований предпочесть одни определения «справедливости», «согласованности», «равновесия» другим. Обоснование принципов выбора решения в этих задачах выходит за рамки математики. Единственное, что могут в таких ситуациях обеспечить формально-логические рассуждения, — это некоторое сужение множества допустимых решений (например, до так называемого множества Парето — множества эффективных точек).

Невозможность или огромная сложность (в любом разумном смысле) четкой формализации ключевых понятий современных разделов теории принятия решений исключает надежные основания для автоматизации соответствующих процессов. Между тем, в повседневной практике, в производственной и общественной жизни необходимость в координации целей, согласовании интересов и разрешении конфликтов встречается на каждом шагу. Люди часто без особого труда неформально согласовывают цели и интересы и к всеобщему удовлетворению разрешают возникающие конфликты. Чем больший жизненный и исторический опыт накоплен принимающим решение, тем, как правило, более естественной оказывается координация целей и действий. Чем мудрее договаривающиеся стороны, тем более устойчивым оказывается согласованное решение.

2. Понятия «справедливость» и «компромисс», «мораль» и «нравственность», как и многие другие естественные человеческие категории, — неформальные понятия. Они представляют собой следствия исторического развития общества,

продукт цивилизации. Агрегирование критериев, согласование интересов и разрешение конфликтов — чисто человеческие функции, исполнение которых требует опыта и мудрости. И то и другое — свойства личности, формирующиеся и развивающиеся в процессе обучения и общественных контактов.

Известный советский генетик академик Н. П. Дубинин пишет [14]: «Вполне очевидно, что если бы детей современного человека с момента их рождения лишить условий современной культуры, они остались бы на уровне наших далеких предков, которые жили десятки тысяч лет назад. Однако дети таких «примитивных» людей в условиях современной культуры... поднялись бы на высоты современного человека».

Можно, конечно, сформулировать аксиоматически естественные требования к понятию «справедливость» и к тем или иным категориям морали. В теории групповых решений, теории многокритериальных задач и теории игр имеются многочисленные интуитивные приемлемые определения понятий справедливости, равновесия, согласования интересов и целей, координации действий, агрегирования критериев (см., напр., [6], [20], [31]). Каждый такой набор аксиом «рационального» агрегирования показателей многокритериальной задачи может быть использован для формирования машинных принципов, в соответствии с которыми искусственный интеллект будет принимать решение при отсутствии объективных четко сформулированных целей. Каждый набор аксиом, определяющий «рациональное» групповое решение, или «справедливый» дележ благ, или «компромиссное» разрешение конфликта может быть принят в качестве формальной основы некоторой машинной морали, устанавливающей правила поведения машинного сообщества — общества интеллектуальных роботов (вспомним, например, нечеткие неоднозначно интерпретируемые азимовские законы робототехники).

Представляется, однако, что любые четко сформулированные однозначно истолковываемые «моральные» заповеди, интуитивно соответствующие исторически сложившимся нормам в одних ситуациях, могут привести к решениям, совершенно неприемлемым для людей в других ситуациях. И совсем неясно, какие из известных аксиом, определяющих «рациональное согласование интересов и целей», полнее отражают в той или иной обстановке реальные нравственные нормы. Больше того, представляется крайне сомнительной

возможность достаточно полного формального описания динамических категорий, отображающих человеческое поведение,— категорий, существенно зависящих от развивающихся условий жизни и общественных отношений. Здесь уместно напомнить известный парадокс Эрроу [47], в котором система аксиом, определяющая понятие справедливого группового решения, приводит к невозможности согласования интересов без диктатора. При этом каждая из аксиом в отдельности представляется вполне интуитивно приемлемой.

В западной печати много пишут о внедрении электроники в быт — об электронной кухне, о роботе-няньке, об автоматической уборщице и т. д. Конечно, многие из функций домохозяйки могут быть формализованы и переданы автоматам. Можно надеяться, что роботы справятся с этими функциями гораздо лучше, чем человек. Но в, казалось бы, рутинной работе домохозяйки могут встретиться неподвижные обстоятельства. В таких ситуациях автомат с электронными мозгами может сварить несъедобный обед, утопить в ванне ребенка, выбросить алмаз в мусорную яму. Чтобы исключить нежелательные решения, нужно наделить робота чувствами, научить его понимать, «что такое хорошо и что такое плохо», отличать добро от зла. Но кто может поручиться, что при этом искусственный интеллект — мозг робота, вкусив плоды от дерева познания, не придет к выводу, что лучше переложить трудоемкие работы на свою хозяйку и заменить ее в более приятных операциях, что не возникнет ситуация, описанная в фантастическом памфлете И. Варшавского «Перпетуум мобиле»:

«— Постойте! — в голосе Пентода звучат радостные нотки.— А почему мы вообще обязаны это делать?

— Что делать?

— Кормить и обслуживать людей.

— Но они же совершенно беспомощны,— растерянно говорит председатель.— Лишить их обслуживания равносильно убийству. Мы не можем быть столь неблагодарны по отношению к нашим бывшим творцам.

— Чепуха! — вмешивается Конденсатор.— Мы научим их делать каменные орудия.

— И обрабатывать ими землю,— радостно добавляет Феррит.— Пожалуй, это выход. Так мы и решим».

Возникает вопрос, много ли найдется охотников жить по соседству с обществом роботов, обладающих синтетическими нормами поведения, железной логикой, выдающими-

ся вычислительными возможностями и способностью воспроизводить себе подобных (последнее уже не фантазия)!? Авторам, например, такая перспектива не совсем по душе.

Мы видим единственную принципиальную возможность наделить вычислительную машину сугубо человеческими понятиями справедливости, мудрости, морали. Это — моделирование процесса развития личности и ее окружения, точнее, моделирование эволюции общества. Однако, как мы увидим несколько ниже, решение этой задачи — если она вообще осмыслена! — беспредельно сложно в любом мыслимом понимании этого слова.

3. Злоупотребление могущественными силами современной науки и техники чревато опасностями. Проблемы, связанные с решением кардинальных вопросов существования человеческого общества, с категориями добра и зла, войны и мира, — сугубо человеческие проблемы, и решение их можно доверить лишь коллективному человеческому разуму.

Читатель вправе здесь предъявить нам претензии. Авторы, мол, запугивают читателя возможными последствиями вмешательства в жизнь общества искусственного интеллекта с синтетическими нормами нравственности. Но ведь и естественный интеллект — человеческий разум — еще не может похвастать большими достижениями в установлении рая на Земле. Мы отвергаем это возражение. Вся история человечества свидетельствует о поступательном общественном прогрессе. История сметает с пути системы, не отвечающие объективным законам развития общества, оттачивая и совершенствуя при этом нравственные нормы общественной жизни.

Трудно придумать механизм, ограничивающий направления развития предоставленного самому себе искусственного интеллекта или общества роботов. Для любителей повернуть ход истории вспять нет более подходящего способа избежать осуждения, чем переложить ответственность за решения на искусственный интеллект, у которого исторически сложившиеся категории — мораль, гуманность, справедливость и др. — подменены теми или иными аксиоматически сформулированными понятиями равновесия, компромисса, устойчивости.

Сложность моделирования моральных категорий

1. Ниже мы сосредоточим внимание на одном из аспектов проявления «сугубо человеческого» — на категориях морали, играющих важнейшую роль в решениях, принимаемых человеком.

Попытаемся оценить минимальный объем информации, подлежащей переработке при моделировании естественного процесса установления моральных норм общественного поведения. Таким образом можно будет получить представление о затратах времени, необходимого для обучения вычислительной машины принятию решений в согласии с формируемой обществом шкалой ценностей, дополняющей свод законов моральным кодексом.

Формирование и эволюция норм социального поведения — исторический процесс. Это значит, что для установления современной шкалы ценностей, принятой в той или иной социальной группе, нельзя ограничиться моделированием решений, принятых одним поколением. Мораль — историческая категория, сохраняющая элементы, присущие прошлому, и эволюционирующая вместе с изменяющимися общественными отношениями. Человек всегда и всюду носит с собой свою индивидуальную историю и историю человечества. Моделирование решений человека как члена общества требует воспроизведения всей истории человеческой мысли. А. Н. Колмогоров говорил, что автомат, способный писать стихи на уровне больших поэтов, нельзя построить проще, чем промоделировав все развитие культурной жизни того общества, в котором поэты реально развиваются. Богатство внутреннего мира человека есть следствие богатства его воспитания, обучения, деятельности и общественных связей. За игнорированием исторических корней человека и общества при выборе ответственных решений неотвратимо последует запись категорий гуманности в «Красную книгу». Таким образом, модель становления нравственных норм должна содержать блоки, отвечающие системам ценностей различных социальных групп разных поколений, и блок, отражающий эволюцию моральных норм во времени. Моделирование должно соответствовать процессу формирования индивидуальных, групповых, классовых и т. п. норм (норм нравственности социальных коллективов, эпохи и цивилизации в целом).

Наметим контуры примитивных блочных иерархических моделей, позволяющих получить грубую оценку снизу

трудоемкости обучения неформализуемым моральным категориям. Все блоки модели конструируются по единому принципу. Элементы блока (люди, социальные группы, поколения) располагают наборами отношений предпочтения, унаследованными от предшествующих поколений. Предпочтения заданы на множествах альтернатив, определяющих поведение элемента. Элементы блока используют ресурсы из общих запасов. Этих ресурсов явно недостаточно для удовлетворения всех потребностей всех элементов блока. Возникает необходимость в корректировании потребностей и предпочтений. Сдвиг предпочтений должен обеспечить согласование групповых решений и выживание цивилизации.

Пусть блок содержит n элементов, каждый из которых может менять свои предпочтения на множествах альтернатив. Предположим, что каждый элемент блока может выбирать свои предпочтения из некоторого множества, содержащего m отношений, и при этом индивидуальные предпочтения в принципе допускают согласование интересов. Таким образом, групповые предпочтения блока формируются из множества в m^n наборов индивидуальных отношений предпочтения элементов блока. Можно предполагать, что процесс совершенствования групповых предпочтений блока — норм поведения элементов — представляет собой процесс типа естественного отбора. Набор индивидуальных предпочтений, не способствующий «выживанию» блока, не может быть включен в групповые предпочтения. Само собой разумеется, что все блоки модели взаимосвязаны (по крайней мере опосредованно) и связи между ними влияют на построение групповых предпочтений.

Вряд ли следует полагать, что формирование устойчивых групповых предпочтений блока потребует перебора всех или почти всех из наборов индивидуальных предпочтений. Но нет также и оснований надеяться на то, что согласование интересов может наступить достаточно быстро. Альтернативы, отвечающие групповым предпочтениям, естественно по крайней мере считать эффективными точками (точками Парето) исходного множества альтернатив, на котором заданы индивидуальные предпочтения. Определение точки Парето множества альтернатив при индивидуальных предпочтениях, не связанных специальными условиями, требует экспоненциального по n перебора. Следовательно, даже моделирование отдельного блока при немалых n , соответствующих реальному составу блока, — неподъем-

ная задача при любом мыслимом быстродействии вычислительной машины. Неизмеримо более сложную задачу представляет собой моделирование эволюции общества, требующее учета взаимосвязей элементов разных блоков.

2. Другое более формальное рассуждение на эту тему, основанное на предложенном Кемени и Снеллом [19] подходе к упорядочению по предпочтениям, приводит к тем же выводам.

Можно согласиться с тем, что «справедливое» групповое решение должно быть наименее отклоняющимся от индивидуальных решений. Все дело, однако, в том, как определить «метрику» в пространстве предпочтений — «расстояние» между групповым и индивидуальными упорядочениями. Кемени и Снелл сформулировали достаточно естественные аксиоматические требования к «метрике» и доказали, что единственное определение расстояния между упорядочениями, задаваемыми матрицами $A^{(1)}$ и $A^{(2)}$, устанавливается функцией

$$d(A^{(1)}, A^{(2)}) = \frac{1}{2} \sum_{i,j} |a_{ij}^{(1)} - a_{ij}^{(2)}|.$$

Здесь $A^{(k)} = \|a_{i,j}^k\|$ — матрица порядка m^n упорядочений предпочтений блока, отвечающая k -му элементу блока. Элементы a_{ij} матрицы A удовлетворяют условиям

(I) $a_{ij} = 1, 0$ или -1 ;

(II) $a_{ij} = -a_{ji}$;

(III) если $a_{ij} \geq 0$ и $a_{jk} \geq 0$, то $a_{ik} \geq 0$, причем $a_{ik} = 0$ в том и только в том случае, если $a_{ij} = a_{jk} = 0$.

Обозначим через D область определения матриц A , элементы которых удовлетворяют условиям (I) — (III).

Выбор матрицы A_r группового решения, наилучшим образом согласующегося с индивидуальными упорядочениями элементов блока, сводится к решению задачи дискретного программирования:

$$\sum_{i=1}^n d(A^{(i)}, A_r) \rightarrow \min | A_r \in D.$$

Приведенное ранее обсуждение вычислительной сложности задач целочисленного программирования не внушает оптимизма по поводу возможности механического установления «справедливых» моральных норм за обозримое время.

В наших рассуждениях не учтены многие факторы, существенно влияющие на установление групповых предпоч-

тений в блоках, и игнорируются затраты времени на получение исходной информации для моделирования. Таким образом, полученная экспоненциальная по размерности системы оценка трудоемкости моделирования отдельного блока может служить лишь нижней оценкой (причем очень грубой) сложности моделирования процесса установления и эволюции моральных категорий общества. Но и она слишком велика для того, чтобы можно было всерьез говорить о возможности построения «справедливой» «машинной морали». Искусственный интеллект может породить лишь искусственную мораль.

3. Рассмотрим более простую задачу о выявлении принципов выбора в той или иной области человеческой деятельности по результатам наблюдений за действиями опытного ЛПР (лица, принимающего решение).

В математической экономике исследуется и обсуждается понятие выявленного предпочтения по данным выборочного обследования. Это понятие лежит в основе теории спроса и используется для разработки требований к плану производства, обеспечивающему удовлетворение платежеспособного спроса. По аналогии с понятием «выявленное предпочтение потребителя» может быть построено более общее понятие «выявленный механизм выбора ЛПР». Идея формирования такого понятия состоит в следующем. В результате наблюдения за поведением ЛПР фиксируются ситуации, требующие выбора, и принятые решения. Соответственно сопоставляются множества альтернатив, отвечающие предъявлениям (ситуациям выбора), и их подмножества — реализованные выборы. Таким образом, могут быть составлены так называемые функции выбора $C(X) \subseteq X$, где X — предъявленное множество альтернатив, $C(X)$ — отобранное множество альтернатив; G — множество всевозможных альтернатив. Во всех случаях, когда удастся построить всюду определенную (на G) функцию выбора, возникает возможность имитировать действия ЛПР. Однако, как правило, множество ситуаций, в которых приходится принимать решение, столь велико, что за обозримое время удастся построить по данным наблюдений далеко не полностью определенную функцию выбора. Тем не менее во многих практических задачах данные ограниченного эксперимента могут служить представительной выборкой, позволяющей выявить закономерности механизма, в соответствии с которым специалист принимает решение, или по крайней мере, высказать правдоподобные гипотезы по

этому поводу. Собственно, именно по такому пути переходят от накопления опыта к построению теории в естественных науках. «Природа коварна, но не злонамеренна». Законы природы устанавливаются на ограниченном опытном материале.

Исследования, проводящиеся в последнее время, позволяют привести в соответствие каждой функции выбора, рассматриваемой как концепция принятия решений, некоторый механизм выбора — так называемую многошаговую схему обобщенного математического программирования (см., напр., [56] и цитируемую в ней литературу). Для произвольной функции выбора число шагов многошагового механизма принятия решений оценивается числом альтернатив в предъявлении и не может быть существенно уменьшено. Однако если функция выбора удовлетворяет некоторым нежестким допущениям, так или иначе характеризующим «рациональный» выбор, можно радикально сократить число шагов механизма и упростить структуру задачи, решаемой на каждом шаге. В таких случаях механизм принятия решений, сконструированный по не полностью определенной функции выбора, может быть с успехом использован и в ситуациях, не наблюдавшихся в эксперименте. Другими словами, если функция выбора удовлетворяет требованиям к «рациональному» выбору, можно построить механизм принятия решений по ограниченному объему наблюдений.

Предварительный анализ методов построения многошаговой схемы обобщенного математического программирования по заданной концепции «рационального» выбора приводит к следующим выводам.

(1) При выполнении некоторых требований к «рациональному» выбору могут быть построены механизмы принятия решения приемлемой трудоемкости, восстанавливающие функцию выбора ЛПР по ограниченному объему наблюдений. Типичная задача, в которой оказывается полезным разработанный аппарат, — это, например, создание схем, существенно облегчающих работу диспетчера, управляющего воздушным движением.

(2) В конфликтных ситуациях «представительная выборка» стратегий опытного «игрока», позволяющая восстановить его поведение в «игре», как правило, мало отличается от множества всевозможных стратегий. Отсюда сложность автоматизации выбора «рациональной» линии поведения в конфликте.

(3) В общем случае выяснение принципиальной возможности конструирования механизма выбора по ограниченной выборке наблюдений за действиями «эталонного» ЛПР представляет собой *NP*-полную задачу.

Таким образом, ни моделирование эволюции общественных отношений, ни наблюдения за решениями лиц, имеющих опыт и знания в соответствующей области, не дают оснований для надежд на формализацию понятия «компромиссный» или «справедливый» выбор в сложных ситуациях. Слишком долго должен обучаться искусственный интеллект, чтобы ему можно было доверить решение стратегических проблем человеческого общества.

4. Здесь уместно сделать следующее замечание.

Отрицая возможность построения приемлемой «машинной» морали, не следует впадать в другую крайность — преувеличивать роль человеческой интуиции.

Аргентинский физик и философ М. Бунге, известный советским читателям своими переведенными на русский язык монографиями «Причинность» (1962) и «Интуиция и наука» (1967), замечает, что нацизм в сфере идеологии был подготовлен многочисленными философами и «духоучеными», возносившими интуицию и противопоставляющими инстинкты толпы, подогреваемой безответственными фюрерами, науке и разуму. А. Колнаи [24], излагая идеологию нацизма и его философских предшественников, пишет: «...народ, доведенный до звероподобного состояния враждебной разуму догмой, легче склонить к совершению иррациональных поступков, чем народ, бдительность которого поддерживается критикой».

В проблемах морали машинный интеллект и инстинкт и интуиция — две стороны одной медали. Искусственного интеллекта и интеллектуальной интуиции при всей их положительной роли в решении сложнейших технических проблем недостаточно для формирования шкалы человеческих ценностей. Как же определить понятия «справедливость», «компромисс», «равновесие» и другие нормы общественного поведения, составляющие основу выбора решений в сложных ситуациях? Ответ на этот вопрос выходит за рамки обсуждаемой здесь проблемы. Ясно лишь, что он невыносим без синтеза общественных и естественных наук. Одними сложностными соображениями, достаточными для того, чтобы аргументировать невозможность формализации моральных норм, здесь не обойтись.

5. Художник и писатель, участник французского Со-

противления Веркор в книге-эссе «Во что я верю» пишет, что эволюция нравов происходит в направлении, прямо противоположном эволюции животного мира. Человечество, вместо того чтобы подчиняться слепым инстинктам, борется с ними, вместо того чтобы предпочитать грубую силу, создает предпосылки для защиты слабых. Таким образом, род человеческий, нарушая природный уклад жизни, освобождался от того, что у него оставалось от животного. Всякая попытка вернуться к беспощадной системе естественного отбора — возродить практику господства сильного над слабым — отбрасывала человечество в далекое прошлое. Такие поползновения повернуть ход истории назад неизменно пресекались прогрессивными силами.

Литература и искусство, этика и мораль также необходимы для эволюции человеческого рода, как и производство и наука. Под влиянием Шекспира и Достоевского, Вольтера и Толстого, Бетховена и Чайковского, Рафаэля и Репина меняется видение мира. Люди преобразовывают конкретную реальность, стремясь создать мир, подобный тому образу, который их покори́л. Искусственный интеллект — верный помощник человека в решении технических и технологических проблем сколь угодно широкого масштаба, может активно способствовать или противодействовать прогрессивной эволюции нравов. Первая альтернатива может иметь место, лишь если искусственный интеллект сумеет впитать в себя многовековую духовную культуру человечества. Насколько сложна эта задача, мы уже видели.

Границы использования искусственного интеллекта

1. Практическая невозможность обучения машины выбору решений в соответствии с исторически сложившимися нормами нравственности налагает определенные ограничения на перспективы использования вычислительной техники. Развитие и применение искусственного интеллекта должно быть ограничено выбором решений в областях человеческой деятельности, не требующих мудрости, не посягающих на условия жизни последующих поколений. Еще Н. Винер в своей последней работе «Творец и робот» предупреждал, что бездумное развитие и использование вычислительной техники грозят порабощением человечества. Винер провозглашает «машине машинное, человеку чело-

веческое». К сожалению, не все деятели, которым «отец кибернетики» адресовал свои предостережения, деятели, определяющие направления развития и применения искусственного интеллекта, разделяют это мнение основоположника кибернетики. Опасность передачи машине противоположных ей функций существует.

Позаимствуем из книги Дж. Вейценбаума [5] высказывания нескольких апологетов глобального использования искусственного интеллекта, которым автор присвоил кличку искусственных интеллигентов. Аналогичные материалы содержатся и в книге Х. Дрейфуса [13].

Директор ведущей американской вычислительной лаборатории (его имя Вейценбаум не называет) в документе, обосновывающем перспективные исследования, говорит, в частности, что «...вычислительная машина включила себя и несомненно будет продолжать включать себя в осуществление большинства функций, имеющих фундаментальное значение для существования, защиты и развития нашего общества...» [5, с. 309]. Предполагается, следовательно, что машине будет доверено, в частности, и решение вопросов добра и зла, войны и мира. Цитируя этот документ, Вейценбаум указывает, что в нем нет и намек на вопрос, хотим ли мы такого будущего. Этот вопрос просто не подлежит обсуждению.

Американский специалист по искусственному интеллекту Г. Саймон в 1960 г. предсказывал: «В очень близком будущем (много раньше, чем через 25 лет) мы добьемся технической возможности возложить на машины все функции, выполняемые человеком в организации» [5].

Профессор Дж. Форрестер, известный советским читателям по его переведенным на русский язык книгам «Мировая динамика» и «Динамика развития городов», идет еще дальше: «Величайшая неопределенность, свойственная мысленным моделям, заключается в невозможности предвидеть последствия взаимодействия отдельных частей системы. Эта неопределенность полностью ликвидирована в машинных моделях» [5].

Специалист по искусственному интеллекту профессор Массачусетского технологического института М. Минский торжественно заявляет: «При жизни нашего поколения... останется лишь немного интеллектуальных задач, которые будут не под силу машинам: проблема создания «искусственного интеллекта» будет в основном решена» [30, с. 2].

Может показаться, что все (или почти все) в этих выска-

зываниях верно. Но это обманчивое впечатление. Оценки сложности процедур передачи функций естественного интеллекта искусственному без ущерба для шкалы человеческих ценностей приводят к однозначному выводу: затраты ресурсов, в частности временных, на безболезненную передачу интеллектуальных, сугубо человеческих функций машине превышают величины, о которых можно серьезно вести речь. Как не вспомнить при этом сакраментальный ответ Ж. Ж. Руссо на вопрос: способствовало ли возрождение наук и искусств улучшению нравов? Говорили ведь в старину: кто успевает в науках, но отстаёт в нравах, тот больше отстаёт, чем успевает.

В цитируемой книге Вейценбаума приводится еще ряд высказываний «искусственных интеллигентов» (в том числе Б. Ф. Скиннера — лидера бихевиористского направления в психологии), которые считают невозможным доверять человеческому мозгу и нечеткому человеческому языку и ощущают необходимость в создании вычислительных машин, «на которые можно полагаться». Цитируя высказывания «мессий от техники» о преимуществах машинного языка над человеческим, Вейценбаум восклицает: «Остается только удивляться, что этот обычный язык... до сих пор существует. И если столь очевидно, что любые понятия и отношения можно перевести в термины машинного языка, то почему же лингвисты, например Халле, Якобсон, Хомский, все еще так упорно бьются над своими проблемами? И почему еще существуют поэты?» [5].

Здесь уместно напомнить, что автор этих слов не «гуманитарий», а «технар». Вейценбаум составил в 60-х годах программу «Элиза», позволяющую имитировать первичное обследование пациента психотерапевтом. Вейценбаума поразило, что квалифицированные психиатры и образованные инженеры всерьез решили, что подобная программа может стать основой автоматизированной психотерапии и, в частности, решает задачу понимания машиной естественного человеческого языка. Тревожное чувство остаться непонятым заставило его взяться за перо.

Изучение категории «сложность» привело нас к тем же выводам совсем по другим соображениям.

2. Повторим еще раз, что мы никоим образом не подвергаем сомнению роль и возможности искусственного интеллекта в исследовании кардинальных проблем естествознания и техники — в анализе любых задач, которые могут быть так или иначе формализованы и сложность решения

которых имеет обозримую оценку. Наши сомнения относятся главным образом к возможности и целесообразности автоматизированного решения таких интеллектуальных задач, как задачи принятия решений, связанные с наличием многих критериев и столкновением различных интересов. Такие задачи содержат неформализуемую компоненту — принцип установления справедливого в данной ситуации компромисса. Это не математическая, а сугубо гуманитарная задача. Основанием для ее решения является исторически сложившаяся шкала общественных ценностей и мудрость лица, принимающего решение. По-видимому, и мы это уже неоднократно подчеркивали, единственный путь формализации таких понятий связан с моделированием развития лица, принимающего решение, и его окружения. Это задача колоссальной сложности.

В цитированных замечаниях специалистов по искусственному интеллекту возлагаются большие надежды на непрерывный рост производительности вычислительной техники. В последнее время много разговоров ведется вокруг японского проекта создания вычислительных комплексов пятого поколения. Для японцев это национальная программа. Для ее решения они сосредоточили свой промышленный потенциал и интеллектуальную элиту. Выделены огромные ассигнования и налаживается международное сотрудничество. Какими только терминами ни величают эту проблему. Предполагается снабдить эту мощную технику знаниями и средствами использования знаний. Правда, в доступной околонаучной литературе не уточняется, что такое человеческие знания, каковы их природа, форма, объем, что такое мыслительные возможности человека. Нам представляется, что по всем этим вопросам мы знаем немногим больше, чем ничего. Но даже оставляя на совести авторов рекламных изданий сверхзадачи проекта, трудно переоценить возможности ЭВМ пятого поколения. А ведь за ним последуют в обозримом будущем и очередные поколения компьютеров. Лидерство в информационной технике безусловно будет содействовать и завоеванию ведущего положения в других областях. Но приведет ли стремительное развитие «индустрии переработки информации» к целесообразности передачи искусственному интеллекту решения проблем справедливого согласования интересов?

Приведенные ранее рассуждения позволяли бы ожидать положительный ответ на эти вопросы, если бы предвидимый рост производительности и «интеллекта» вычислительной

техники обеспечил моделирование общественных черт эволюции поведения человека и общества. Мы видели, однако, что даже самые простые и грубые подходы к формализации понятия «справедливое согласование интересов» сводятся к решению задач переборного типа. Имеются основания считать эти задачи экспоненциально сложными. Реальный механизм эволюции жизни за миллионы лет на миллиардах организмов отшлифовал пусть еще далеко не совершенные понятия морали, позволяющие уживаться при ограниченных ресурсах огромным коллективам людей. «Повторить» этот процесс за обозримое время при любом мыслимом ускорении (определяемом даже потенциальными возможностями вычислительной техники) — нереальная задача. Таблицы 2 и 3, оценивающие влияние производительности ЭВМ на размерность разрешимой задачи и максимальную размерность задач, разрешимых за заданное время, иллюстрируют этот вывод для задач экспоненциальной сложности.

Таким образом, в задачах принятия решений, в которых не обойтись без понятия «справедливый компромисс», искусственный интеллект и в обозримом будущем будет нуждаться в участии естественного интеллекта. Совершенствование выбираемых решений нуждается не столько в усилении искусственного интеллекта, сколько в разработке эффективных путей взаимодействия искусственного и естественного интеллекта. Рациональный симбиоз человека и машины, в котором человеку отводится человеческое, а машине — машинное, — вот основное направление, способствующее принятию решений, требующих согласования интересов.

Преувеличение возможностей искусственного интеллекта в выборе компромиссных решений имеет те же корни, что и неоправданные надежды на глобальную автоматизацию в первые годы кибернетического бума. И то и другое — следствия определенных успехов, достигнутых кибернетиками и специалистами по искусственному интеллекту при решении относительно простых задач. Выводы, полученные при анализе поведения простых автоматов в несложных средах и «умных» машин, принимающих рациональные решения в «эталонных» ситуациях, с малозначащими оговорками экстраполировались безответственными популяризаторами на сколь угодно сложные системы и произвольные ситуации.

Реальные явления и процессы много сложнее кибернетических игрушек, а выбор рациональной стратегии в сколь-

нибудь серьезной конфликтной ситуации не сравнить по трудоемкости с автоматизированным выбором лучшего хода в игре «крестики и нолики». Теория сложности имеет веские основания утверждать, что в подавляющем большинстве классов задач принятия решений трудоемкость наиболее экономных методов (например, число итераций или элементарных действий, необходимых для выбора решения требуемой точности) растет неизмеримо быстрее «сложности» задач класса (например, ее размерности — числа параметров, определяющих решение). Представляется, что именно поэтому выдающиеся пропагандисты ультрановых направлений сменили со временем первоначальные восторги на нотки разочарования. Так, последняя книга «отца кибернетики» Норберта Винера «Творец и робот», впрочем, как и «Новые главы кибернетики», написаны совсем не в той тональности, что «Кибернетика» или «Кибернетика и общество».

Стоит также обратить внимание на то, какую эволюцию прошел видный ученый и талантливый писатель-фантаст Айзек Азимов, сменивший благоговение перед сверхинтеллектом мозга-гиганта («Три закона робототехники») на откровенную насмешку над электронным разумом Мультивака («Машина-победитель»).

Вице-президент АН СССР академик Е. П. Велихов в предисловии к переводу предварительного отчета Японского комитета научных исследований в области создания ЭВМ пятого поколения [44] говорит об ограниченной «способности» ЭВМ к анализу вновь поступившей информации и о неумении вычислительной машины с достаточной степенью надежности выявить содержательность вновь поступивших данных. Е. П. Велихов замечает: «Возникающие в связи с этим препятствия, вероятно, принципиально **непреодолимы**». И далее: «...искусство выбирать решения в подлинно сложной обстановке еще долгое время будет недоступным компьютеру, который лишен способности действовать субъективно, соизмеряя решения и поступки с мерой ответственности и иными важными для человека атрибутами существования» [44, с. 9].

Возможно ли единое определение категории «сложность»?

Мы исследовали различные компоненты сложности — интуитивно приемлемые характеристики трудоемкости методов решения массовых проблем и минимальных ресурсов, гарантирующих требуемое качество описания или синтеза систем заданных классов. Изучение этих составляющих категории «сложность» позволяет в ряде случаев (исходя из познанных логических, физических и биологических закономерностей) оценить количественно или по крайней мере качественно потенциальные возможности целенаправленных действий и процессов и содействовать, таким образом, прогнозированию будущего.

Мы с самого начала подчеркивали, что на современном уровне знаний понятие «сложность» вряд ли можно рассматривать как скалярное, что оно является многогранным — векторным — понятием и точный смысл его определяется в соответствии с конкретными особенностями изучаемых ситуаций и проблем. Отсутствие единого определения категории «сложность» оставляет, однако, чувство неудовлетворенности. Ведь до К. Шеннона и понятие «количество информации» считалось многозначным. Его точный смысл обуславливался спецификой передаваемой или хранимой информации. А. Н. Колмогоров отмечает [21], что по первоначальному замыслу понятие «информации» не связывалось со скалярной величиной. «Различные виды информации могут быть чрезвычайно разнообразны. Можно было заранее ожидать, что можно предложить те или иные способы измерять «количество» информации, но было не ясно, существует ли среди этих способов какой-либо один, имеющий принципиальное преимущество перед другими, а главное, было совершенно не ясно, можно ли качественно различить информации, для которых та или иная условная количественная мера одинакова, действительно считать эквивалентными в смысле трудности их передачи через каналы связи или их хранения в запоминающих устройствах. Оказалось, что такая «правильная» по преимуществу мера количества информации существует и позволяет решать до конца широкий класс задач, в которых априори независимость решения от более тонких качественных особенностей информации была совершенно не ясна».

Выдающийся вклад Шеннона заключается, в частности, в том, что он сумел сформулировать определение «количества информации» как единственную скалярную функцию, удовлетворяющую некоторым разумным аксиомам [43] — естественным требованиям к численной оценке объема передаваемой или хранимой информации. Определение Шеннона удовлетворило всех потребителей этого понятия во всех ситуациях, где оно использовалось.

Хотелось бы надеяться, что по мере роста понимания природы сложности будет предложено и единое определение категории «сложность», так что формулировка каждой из ее существенных компонент будет при некоторых предположениях вытекать из общего определения как частный случай. Не следует, однако, себя чрезмерно обольщать по этому поводу и закрывать глаза на трудности, связанные с такой задачей. Понятие количества информации относится к физике. Категория же сложности должна обслуживать весь спектр проблем, относящихся не только к разделам естествознания, но и к разным аспектам человеческой деятельности. В физике мы работаем с относительно простыми системами. Обращаясь к наукам, лежащим за пределами физики, мы сталкиваемся с более сложными (в любом разумном интуитивном смысле) структурами, с большим числом переменных и взаимосвязей, с более серьезными и менее случайными нелинейными эффектами. Мы еще не располагаем достаточно общими мерами сложности таких структур.

При использовании математических методов в физике наличие обоснованных физических принципов, устанавливающих количественную связь между основными физическими понятиями, позволяет измерять эффективность физических процессов. В физике имеются фундаментальные физические постоянные, устанавливающие пределы осуществимости тех или иных физических процессов и характеризующие затраты, связанные с их реализацией. Не так обстоит дело в других науках. Как измерить сложность биологических, экономических систем, как оценить эффективность решения задач в той или иной области?

Приемлемое общее определение категории сложности, если оно вообще возможно, породило бы надежду на возможность установления общих закономерностей поведения сложных систем, не зависящих от их природы, и получения оценок достижимости тех или иных целей при различных подходах к управлению сложными системами. Математика,

по словам А. Пуанкаре, — это искусство давать разным вещам одно название. Изучив одну закономерность, исследователь узнает ее затем во множестве самых различных ситуаций. По-видимому, это наиболее естественный путь познания закономерностей все усложняющейся действительности.

Пути преодоления сложности

1. Сложность явлений и процессов зависит не только от их объективных характеристик, но и от уровня наших знаний о них и от выбранного способа описания изучаемых объектов. В средние века умение производить арифметические операции было признаком большой учености, в настоящее время с этим справляются ученики младших классов. В системе Птолемея планеты двигались по сложным орбитам, в системе Коперника описание орбит предельно просто.

Компоненты сложности, исследованные и формализованные в предыдущих главах, обсуждались в понятиях и терминах логики и вычислительной математики. Явления и процессы, сложностные характеристики которых оценивались, разлагались на элементарные составляющие. Представления о целом основывались на изучении элементов и взаимосвязей между ними. Таков основной, но не единственный подход математики к познанию реальности. Это надежный, доказательный, строгий подход. Но умение логически мыслить, умение доказывать — это еще не вся математика. Установление качественных и количественных закономерностей требует, кроме того, и целостного взгляда на процессы и явления. Математика как инструмент познания полезна, когда удастся концентрировать внимание на центральных вопросах и игнорировать второстепенные факторы. Выделению главного способствуют опыт, содержательные знания, интуиция и изобретательность. Логика и вычислительный аппарат становятся инструментами познания, если они дополняются интуицией. Интуиция, конечно, не обладает строгостью и достоверностью логики и не способна обеспечить оценки, которые можно получить с помощью вычислений. Зато интуиция позволяет посмотреть на исследуемое явление издали, откуда хоть и не видны детали, но различимы общие контуры явления и доступны оценки его сглаженных характеристик.

Аналогии, присущие интуитивно-содержательному мышлению, обеспечивают возможность перенести приемы, испытанные при изучении одних задач, на другие. Но как выбрать исходную позицию и угол зрения, позволяющие окинуть взглядом явление в целом, как подметить сходство между вновь возникшей проблемой и изученными ранее? Формально-логического ответа на этот вопрос мы не знаем. Интуитивные возможности исследователя, конечно, возрастают с накоплением опыта. Следует, однако, полагать, что способности к логике или интуиции (особенно к последней) являются врожденными.

2. Логика и интуиция в науке вообще и в математике в частности играют каждая свою необходимую роль. Логика, гарантирующая достоверность, есть орудие доказательства, а интуиция — способность к целостному восприятию — орудие изобретательства. Оба подхода одинаково необходимы для прогресса науки. Отказ от любого из них затуманивает качественные закономерности явления, огрубляет количественные оценки его характеристик.

Возможности объяснения «сложного» через изучение «простого», по-видимому, имеют свои границы. Вряд ли целесообразно объяснять закономерности макромира в терминах и понятиях теории элементарных частиц. Для этого нужны и другие понятия и другие термины, связанные с характеристиками макроявления. Так и объективное исследование категории сложности не может ограничиться логическим анализом компонент сложности, основанным на изучении возможных конфигураций и взаимодействий элементов, составляющих оцениваемый объект. Требуется еще привлечение новых понятий, обеспечивающих целостный взгляд на изучаемое явление и исключающих, таким образом, учет нереализуемых или крайне редко реализуемых в практических условиях конфигураций или взаимодействий компонент изучаемого явления.

Строгий язык алгоритмов и машин оперирует элементарными понятиями. Язык интуиции — язык образов — обладает гибкой динамической семантикой. В емких интуитивных терминах целостного макроописания объекта потери в точности описания с лихвой компенсируются снижением сложности объектов, «аппроксимирующих» исследуемые. Таким образом, на оценки компонент сложности, полученные в предыдущих главах на основе формально-логического подхода, можно смотреть как на оценки сверху

значений сложности, учитывающих, помимо того, интуитивный подход к описанию объекта.

3. Мы пришли к выводу, что путь к преодолению сложности — это объединение формально-логического подхода к описанию и изучению объекта с интуитивным творческим подходом.

Что же такое математическое творчество? Выдающийся французский математик Анри Пуанкаре говорил, что творить в математике — значит, отличать, выбирать. Математическое творчество состоит в комбинировании известных и построении новых понятий, позволяющих установить родство между фактами из удаленных друг от друга областей. Из комбинаций известных математических объектов ничтожное меньшинство оказывается полезным. Выделение таких комбинаций требует колоссального перебора. Поддающаяся контролю логическая деятельность вряд ли способна на это. Подсознание играет в математическом творчестве первостепенную роль. О механизме подсознательного мышления известно очень мало. Как объяснить интуитивный выбор из множества комбинаций понятий и объектов именно тех, которые представляют интерес и стимулируют сознательное мышление? В чем причина того, что из подсознательного хаоса в мозгу одним продуктам деятельности мозга удастся превысить порог сознания, а другим нет?

Можно думать, что так же, как среди всех раздражений наших органов чувств только самые сильные останавливают на себе внимание, так и среди неосознаваемых мозговых процессов подготовлены к осознанию те, которые оказывают наибольшее влияние на нашу способность к восприятию. Среди математиков бытует мнение, что одним из основных критериев отбора при переходе от подсознания к сознанию, от внелогического к логическому мышлению являются эстетические эмоции — чувство гармонии чисел и изящества форм. Математические объекты вызывают ощущение прекрасного, если их структура столь гармонична, что ум способен охватить целое, проникая в то же время в детали. Для математика самые абстрактные сущности являются как бы живыми существами, обладающими каким-то внутренним единством, изящными формами, стройной структурой. Люди творческого труда знакомы с ощущением счастья и радости в моменты озарения, когда после длительных, порой мучительных переживаний, вызванных нерешенной проблемой, вдруг при самых неподходящих обстоятельствах возникает решение задачи. Стимулом к твор-

честву является присущая человеку внутренняя потребность к преодолению трудностей.

Время, в течение которого подсознательно вынашивается подход к решению, может быть достаточно большим. Перебор комбинаций, выделяющий надпороговые идеи, озаряющие мозг, должен быть необычайно велик. Каковы бы ни были осмысленные пределы скорости подсознательного перебора, трудно представить себе случайный выбор комбинации фактов и понятий, составляющих сущность догадки, за обозримое время. Математикам известно, что случайный поиск может быть эффективен только тогда, когда «мощность» множества удачных решений не мала по сравнению с объемом области всех возможных решений. Чтобы повысить эффективность случайного поиска, им следует управлять. Имеются основания ожидать, что интуитивный и сознательный процессы выбора решений тесно переплетаются и представляют собой итеративную последовательность логических и подсознательных этапов. Плодотворной подсознательной деятельности предшествует этап логических рассуждений, сужающих область поиска. Границы интуитивной активности выбираются не наугад. С другой стороны, подсознательная деятельность, как правило, не доставляет готовый результат, а придает некоторый импульс сознанию, которое, в свою очередь, упорядочивает процесс интуитивного поиска в состоянии кажущегося покоя мозга. Конечно, приведенные рассуждения не подкреплены сколько-нибудь серьезной экспериментальной проверкой, но они представляют собой вполне правдоподобную рабочую гипотезу. Эти соображения могут быть использованы не только для постановки работ по изучению физиологии творческой деятельности, но и для разработки техники и технологии «усилителей» интеллекта.

4. Необходимость внелогического этапа — интуиции — в математическом мышлении вытекает и из сугубо формальных соображений. Усилия, предпринятые математиками, чтобы очистить научную систему от интуитивного элемента, привели к противоположному результату.

В так называемой «чистой» математике существуют разделы, все утверждения которых могут быть логически выведены из некоторой системы независимых аксиом. Однако если речь идет о прикладной математике, претендующей на связь с внешним миром, полностью дедуктивная теория невозможна. Это следует из уже упомянутой во второй главе теоремы Геделя, содержательная интерпретация которой

может быть изложена следующим образом. Какова бы ни была независимая, непротиворечивая система аксиом, всегда существуют высказывания, истинность или ложность которых не может быть установлена на основании исходных аксиом. Это значит, что для того, чтобы соответствующая формальная теория справлялась с оценкой всех относящихся к ней утверждений и фактов, необходимо дополнять ее логические основания новыми аксиомами, не зависящими от исходных и не противоречащими им. Другими словами, развитие формальной теории требует непрерывного обогащения ее аксиоматических основ внелогическими интуитивными элементами. Эту интерпретацию теоремы Геделя предвидел Козьма Прутков и выразил ее следующими словами: «Многие вещи нам не понятны не потому, что наши понятия слабы; а потому, что сии вещи не входят в круг наших понятий» (Плоды раздумья, с. 66).

Во многих разделах прикладной математики необходимость в интуитивном расширении логических основ теории возникает на каждом шагу. Например, в многокритериальных задачах, в теории групповых решений, в теории игр каждый новый класс ситуаций требует определения понятия равновесия посредством новых аксиом, отражающих интуитивные представления лица, принимающего решение о справедливом компромиссе. Еще сильнее переплетаются формальные и интуитивные начала в инженерном творчестве.

5. Человечество познает мир двумя путями — с помощью эксперимента и логических рассуждений, с одной стороны, и с помощью чувств и воображения — с другой. Первый путь присущ в большей степени науке, второй — искусству. Наука оперирует понятиями, искусство — образами. Цивилизация обязана своими достижениями не только успехам науки, но и великим творениям искусства. Искусство утверждает авторитет интуитивного постижения и нарушает монополию логического мышления. Искусство способно давать целостное восприятие действительности. Оно противостоит формальному подходу к изучению реальности, когда разные качества и свойства объекта исследуются разными науками и в пределах каждой из них основным методом исследования предполагается разложение целого на простейшие элементы. Хотя, по словам Ояра Ваццетиса, «наука это тоже искусство, если она действительно наука, а не ремесло», исторически сложилось так, что наука и искусство оказались в разных областях человеческой культуры.

Не в синтезе ли логического и ~~гносе~~гносеологического подходов к познанию действительности — основной путь и преодолению сложности изучения окружающего мира? Положительный ответ на этот вопрос не противоречит современным тенденциям в науковедении и искусствознании. Не противостояние «физиков» и «лириков», не противоборство «технофилов» и «технофобов», а их координированные усилия способствуют преодолению сложности познания природы и общества. Ясно, что, какими путями ни пойдет развитие человеческой культуры, за преодоление сложности придется расплачиваться. Возникнут новые понятия, появятся новые объекты исследования, усложнится окружающий мир. Таково непрерывное развитие жизни. Как говорится в прекрасных стихах Эмиля Верхарна («Невозможное»):

Что виделось вчера как цель глазам твоим,—
Для завтрашнего дня — оковы;
Мысль — только пища мыслей новых,
Но голод их неутолим.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Бир Ст. Мы и сложность современного мира.— В кн.: Кибернетика сегодня: проблемы и суждения.— М.: Знание, 1976.
2. Бирюков Б. В., Тютин В. С. О понятии «сложность».— В кн.: Логика и методология науки.— М.: Наука, 1967.
3. Bremerman H. I. Limits of Genetic Control.— IEEE Transactions of Military Electronics, V M.T.L.— 7, № 2, 3, April — July, 1963.
4. Бузыцкий П. Л., Фрейман Г. А. О применении аналитических методов в комбинаторных задачах.— Изв. АН СССР. Техн. кибернетика, 1980, № 2.
5. Вейценбаум Дж. Возможности вычислительных машин и человеческий разум.— М.: Радио и связь, 1982.
6. Вилкас Э. Й., Майминас Е. З. Решения: теория, информация, моделирование.— М.: Радио и связь, 1981.
7. Винер Н. Кибернетика и общество.— М.: ИЛ, 1958.
8. Винер Н. Творец и робот.— М.: ИЛ, 1960.
9. Витушкин А. Г. Оценка сложности задачи табулирования.— М.: Физматгиз, 1959.
10. Генс Г. В., Левнер Е. В. Дискретные оптимизационные задачи и эффективные приближенные алгоритмы. Обзор.— Изв. АН СССР. Техн. кибернетика, 1979, № 6.
11. Гимади Э. Х., Глебов Н. И., Перепелица В. А. Алгоритмы с оценкой для задач дискретной оптимизации. — Проблемы кибернетики. Вып. 31.— М.: Наука, 19
12. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи.— М.: Мир, 1982.
13. Дрейфус Х. Чего не могут вычислительные машины.— М.: Прогресс, 1978.
14. Дубинин Н. П. Вечное движение.— М.: Прогресс, 1973.
15. Заде Л. Основы нового подхода к анализу сложных систем и процессов принятия решений.— В кн.: Математика сегодня.— М.: Знание, 1974.
16. Звонкин А. К., Левин Л. А. Сложность объектов и обоснование понятия информации и случайности с помощью теории алгоритмов.— УМН, 1970, т. 25, № 6.
17. Карп Р. М. Сводимость комбинаторных проблем.— Кибернетический сборник. Вып. 12.— М.: Мир, 1975.
18. Касты Дж. Большие системы. Связность, сложность, катастрофы.— М.: Мир, 1982.
19. Кемени Дж., Снелл Дж. Кибернетическое моделирование.— М.: Советское радио, 1972.

20. Кини Р. Л., Райфа Х. Принятие решений при многих критериях: предпочтения и замещения.— М.: Радио и связь, 1981
21. Колмогоров А. Н. Теория передачи информации.— М.: Изд-во АН СССР, 1956.
22. Колмогоров А. Н. Три подхода к определению понятия «количество информации».— Проблемы передачи информации, 1965, т. I, № 1.
23. Колмогоров А. Н. Комбинаторные основания теории информации и исчисления вероятностей.— УМН, 1983, т. 38, № 4.
24. Kolnai A. The Ware Against the West.— London, Gollanez, 1938.
25. Кук С. А. Сложность процедур вывода теорем.— Кибернетический сборник. Вып. 12.— М.: Мир, 1975.
26. Кульбак С. Теория информации и статистика.— М.: Наука, 1967.
27. Левин Л. А. Универсальные задачи перебора.— Проблемы передачи информации, 1973, т. 9, № 3.
28. Манин Ю. И. Вычислимое и невычислимое.— М.: Советское радио, 1980.
29. Марков А. А. Теория алгорифмов.— Труды Математического института АН СССР, 42, 1954.
30. Minsky M. Computation: Finite and Infinite Machines.— Englewood Cliffs, N. Y. Prentice — Hall, 1967.
31. Миркин Б. Г. Проблема группового выбора.— М.: Наука, 1974.
32. Нейман Дж. фон. Общая и логическая теория автоматов.— В кн.: Тьюринг А. М. Может ли машина мыслить? — М.: Физматгиз, 1960.
33. Немировский А. С., Юдин Д. Б. Сложность задач и эффективность методов оптимизации.— М.: Наука, 1979.
34. Немировский А. С., Юдин Д. Б. Адаптивное управление экономикой.— М.: ЦЭМИ АН СССР, 1982.
35. Немировский А. С., Юдин Д. Б. Информационная сложность математического программирования.— Изв. АН СССР. Техн. кибернетика, 1983, № 1.
36. Пуанкаре А. О науке.— М.: Наука, 1983.
37. Тарьян Р. Э. Сложность комбинаторных алгоритмов.— Кибернетический сборник. Вып. 17.— М.: Мир, 1980.
38. Трахтенброт Б. А. Сложность алгоритмов и вычислений.— Новосибирск: Изд-во НГУ, 1967.
39. Финкельштейн Ю. Ю. Приближенные методы и прикладные задачи дискретного программирования.— М.: Наука, 1976.
40. Хачиян Л. Г. Полиномиальные алгоритмы в линейном программировании.— ЖВМ и МФ, 1980, т. 20, № 1.
41. Хинчин А. Я. Понятие энтропии в теории вероятностей.— УМН, 1953, т. 8, № 3 (55).
42. Chaitin G. I. Toward a Mathematical Definition of «Life».— Symposium MFCS, High Tatras, 1975.
43. Шеннон К. Математическая теория связи.— М.: 1948.
44. ЭВМ пятого поколения. Концепции, проблемы, перспективы / Под ред. Т. Мото-ока.— М.: Финансы и статистика, 1984.
45. Edmonds J. Paths, Trees and Flowers.— Canad. J. Math., 17, 1965.
46. Эйген М. Самоорганизация материи и эволюция биологических макромолекул.— М.: Мир, 1973.

47. Arrow K. J. Social Choice and individual Values,— Wiley, N.Y., 1951.

48. Эшби У. Р. Введение в кибернетику.— М.: ИЛ, 1959.

49. Юдин А. Д. Информативные структуры многомерных случайных величин.— Изв. АН СССР. Техн. кибернетика, 1977, № 6.

50. Юдин А. Д. Асимптотически оптимальный алгоритм решения обобщенной задачи о соединении городов.— Изв. АН СССР. Техн. кибернетика, 1981, № 2.

51. Юдин А. Д. Сложность оценивания статистических систем.— Изв. АН СССР. Техн. кибернетика, 1981, № 6.

52. Юдин А. Д. Сложность статистических систем.— ДАН СССР, 1982, т. 266, № 5.

53. Юдин А. Д. Информативная сложность статистических систем.— Изв. АН СССР. Техн. кибернетика, 1982, № 3.

54. Юдин Д. Б., Немировский А. С. Эффективность случайного поиска в задачах управления.— Изв. АН СССР. Техн. кибернетика 1977, № 3.

55. Юдин Д. Б., Горяшко А. П., Немировский А. С. Математические методы оптимизации устройств и алгоритмов в АСУ.— М.: Радио и связь, 1982.

56. Юдин Д. Б., Шоломов Л. А. Обобщенное математическое программирование и функции выбора.— Известия АН СССР. Техн. кибернетика, 1985, № 3.

57. Юдин Д. Б., Юдин А. Д. Экстремальные модели в экономике. — М.: Экономика, 1979.

СЛОВАРЬ ТЕРМИНОВ

Адаптивная система — система, приспособливающая свое поведение к условиям меняющейся среды.

Алгоритм — правило, определяющее порядок выполнения некоторой совокупности операций для решения задач определенного класса. Эквивалентные между собой формальные уточнения понятия «алгоритм» формулируются в терминах универсальных машин или частично-рекурсивных функций.

Алгоритмическая сложность $K_{\Phi}(y|x)$ (или условная сложность объекта y относительно объекта x) применительно к универсальной машине со структурой $\Phi(p, x)$ — это минимальная длина программы p , для которой $\Phi(p, x) = y$. Если раз и навсегда зафиксировать объект x , то $K_{\Phi}(y|x) = K_{\Phi}(y)$ — сложность объекта y .

Диалоговое программирование — формальная методология организации рационального взаимодействия специалистов различного профиля (математиков и прикладников), совместно участвующих в разработке целенаправленной системы.

Информационная сложность решения задач данного класса с требуемой точностью — это минимальное число обращений к оракулу, при котором еще существует метод решения любой задачи класса с погрешностью, не превышающей заданную.

Класс P переборных задач — класс переборных задач, полиномиально разрешимых на детерминированной машине Тьюринга.

Класс NP переборных задач (NP -полиные задачи) — класс задач, полиномиально разрешимых на недетерминированной машине Тьюринга.

Машина Тьюринга — идеальная вычислительная машина с неограниченными ресурсами, используемая для изучения вычислительных процессов и формального уточнения понятия «алгоритм».

Оракул — источник информации об индивидуальных задачах класса. Определяется множеством допустимых вопросов к нему и множеством возможных ответов. В практических ситуациях оракул — эксперт, имитационная модель или испытательный стенд.

Переборная задача — массовая задача, каждая реализация (индивидуальная задача) которой имеет конечное число вариантов решения и может быть решена (по крайней мере в принципе) с помощью полного перебора вариантов.

Полиномиально сложная задача — переборная задача, вычислительная сложность которой растет как полином от размерности задачи.

Рекурсия — способ определения значения функции в некоторой точке через значения этой функции в конечном числе других точек, которые в каком-то смысле «предшествуют» ей.

Сигнализирующие вычисления $\mu_A(x)$ — характеристики вычислительного процесса, указывающие затраты ресурсов (времени, памяти и

др.) при решении задачи x с помощью алгоритма A . В зависимости от того, какие ресурсы лимитируемы, та или иная сигнализирующая рассматривается как **вычислительная сложность** задачи.

Сложность оценивания класса статистических систем — это минимальное число независимых одинаково распределенных реализаций случайной величины, моделирующей систему, необходимых для того, чтобы с заданной точностью и требуемой надежностью восстановить совместную функцию распределения вероятностей компонент модели любой системы класса.

Статистическая система — система, подходящей моделью поведения которой является многомерная случайная величина.

Универсальная переборная задача — переборная задача, из полиномиальной разрешимости которой следует полиномиальная разрешимость всех других переборных задач.

Экспоненциально сложная задача — переборная задача, вычислительная сложность которой растет как экспонента от размерности задачи.

Энтропия — мера неопределенности ситуации. Может быть использована как мажоранта сложности ситуации и как оценка количества информации, необходимого для восстановления ситуации.

СОДЕРЖАНИЕ

Предисловие	3
Что такое категория «сложность» и зачем она нужна?	5
Интuitивные определения сложности	5
Сложность и неформальные проблемы	7
Математические задачи теории сложности	8
Сложность и прикладные проблемы	10
Возможно ли путешествие по телеграфу?	14
Алгоритмическая сложность	17
Что такое алгоритм?	17
Частично-рекурсивные функции	19
Машина Тьюринга	23
Определение алгоритмической сложности	26
Свойства алгоритмической сложности	30
Сложность объекта	33
Сложность и случайность	35
Алгоритмическая сложность и теория информации	40
Сложность табулирования	45
Сложность и закон необходимого разнообразия	50
Влияние теории алгоритмической сложности на математические дисциплины	56
Предвидимые приложения теории алгоритмической сложности	61
Вычислительная сложность	64
Меры сложности вычислений	64
Инвариантные результаты теории вычислений	69
Классификация по сложности задач дискретной оптимизации	71
Легкорешаемые дискретные задачи	77
Вычислительная сложность переборных задач	84
Полиномиальная разрешимость задач линейного программирования	90
Труднорешаемые дискретные задачи	93
Заключение	102
Информационная сложность	105
Сложность оптимизации	105
Постановка задачи и основные определения	107
Информационная сложность нелинейного (невыпуклого) программирования	110
Информационная сложность выпуклого программирования	112
Субоптимальные законы управления	118
Адаптивное управление экономикой	121
Диалоговое программирование	127
Сложность статистической обработки информации	137
Характеристики сложности статистических систем	137

Сложность оценивания	143
Информативная сложность	149
Выводы	155
Категория «сложность» и искусственный интеллект	156
Проблема «человек — машина»	156
Объективные и субъективные факторы в принятии решений	160
Сложность моделирования моральных категорий	165
Границы использования искусственного интеллекта	171
Заключение	177
Возможно ли единое определение категории «сложность»?	177
Пути преодоления сложности	179
Использованная литература	185
Словарь терминов	188

Давид Борисович ЮДИН
Александр Давидович ЮДИН
ЧИСЛО И МЫСЛЬ
Выпуск 8
(Математики измеряют сложность)

Главный отраслевой редактор А. Нелюбов
Редактор Н. Феоктистова
Мл. редактор Н. Карячкина
Художественный редактор М. Бабичева
Технический редактор А. Красавина
Корректор С. Ткаченко
ИБ № 7049

Сдано в набор 31.01.85. Подписано к печати 07.06.85. А09339.
Формат бумаги 84×108/32. Бумага тип № 3. Гарнитура литературная. Печать высокая. Усл. печ. л. 10,08. Усл. кр.-отт. 10,50. Уч.-изд. л. 10,43. Тираж 40000 экз. Заказ 5-987.
Цена 55 коп. Издательство «Знание» 101835, ГСП, Москва.
Центр проезд Серова, д. 4. Индекс заказа 856713. Отпечатано с матриц 1-й Образцовой типографии имени Жданова г. Москвы на Головном предприятии республиканского производственного объединения «Полиграфкинг», 252057, Киев, ул. Довженко, 3.

