

## **Стохастическая оптимизация и системное программирование<sup>1</sup>**

*О. Н. Граничин, д. ф.-м. н.<sup>2</sup>*

*Санкт-Петербургский государственный университет*

*Oleg\_granichin@mail.ru*

---

Один из возможных способов дословного перевода термина “системное программирование” с английского языка — организация работы системы. Особые теоретические трудности возникают при исследовании сложных систем. Во многих практических задачах механический перенос на сложные системы естественных подходов, работоспособных для простых, приводит к неразрешимым противоречиям. В частности, возникает огромное число, так называемых, неразрешимых задач (за реальное время).

В последнее время все чаще используются новые “рандомизированные” подходы к решению задач организации работы сложных систем, которые дают удовлетворительные ответы с высокой степенью вероятности. В этой статье будет рассказано о некоторых новых результатах в этой области, полученных в последние годы студентами и аспирантами кафедры системного программирования математико-механического факультета СПбГУ.

*Ключевые слова:* рандомизированные алгоритмы, метод Монте-Карло, оптимизация и оценивание, распознавание образов, устойчивая кластеризация, compressive sensing,  $\ell_1$ -оптимизация, рандомизированные измерения.

### **1. Введение**

Один из возможных способов дословного перевода термина “системное программирование” с английского языка — организация работы системы. Особые теоретические трудности возникают при исследовании сложных систем. Во многих практических задачах механический перенос на сложные системы естественных подходов, работоспособных для простых систем, приводит к неразрешимым противоречиям. В частности, возникает огромное число, так называемых, неразрешимых задач (за реальное время).

В последнее время все чаще используются новые “рандомизированные” подходы к решению задач организации работы сложных систем, которые дают удовлетворительные ответы с высокой степенью вероятности. На кафедре системного программирования

---

<sup>1</sup>Работа выполнена при частичной финансовой поддержке ФЦП “Кадры” (госконтракт №16.740.11.0042 от 01.09.2010).

<sup>2</sup>©О. Н. Граничин, 2010

математико-механического факультета СПбГУ эти направления развиваются сотрудниками, аспирантами и студентами под руководством профессора Граничина Олега Николаевича. Новые научные результаты регулярно публикуются как в ведущих российских и международных научных журналах и материалах конференций, так и в специализированном журнале “Стохастическая оптимизация в информатике” ([www.math.spbu.ru/user/gran/stochopt.html](http://www.math.spbu.ru/user/gran/stochopt.html)), ежегодно публикуемом НИИ ИТ в Издательстве Санкт-Петербургского государственного университета и включенном в Российский индекс научного цитирования (РИНЦ, [www.elibrary.ru](http://www.elibrary.ru)).

Традиционный подход к процессу конструирования самых разнообразных систем и алгоритмов управления представляет собой выполнение детерминированных алгоритмов, состоящих из последовательности детерминированных шагов. Этот подход, однако, может быть обобщен включением *рандомизации*. В рандомизированных алгоритмах один или несколько шагов основываются на случайном правиле, при котором среди многих детерминированных правил одно выбирается в соответствии с некоторой случайной схемой. Рандомизация позволяет ввести новый термин “вероятностно успешный алгоритм”. Во многих случаях, когда детерминистической успешности достичь невозможно, вероятностная успешность предлагается как значимая альтернатива. Рандомизация становится мощным средством для решения целого ряда задач организации работы сложных систем, считающихся неразрешимыми с помощью детерминированных методов.

В этой статье будут кратко описаны общие идеи теории рандомизированного обучения (Randomized learning theory), оптимизации функционалов среднего риска (mean-risk), новые принципы функционирования будущих вычислительных устройств, описываемых как гибридные системы наборов асинхронных стохастических динамических систем, указаны дипломные работы студентов по рандомизированным методам обработки изображений, видео, информации о финансовых данных, мультиагентному управлению и беспилотным летательным аппаратам.

## 2. Теория рандомизированного обучения

Большое количество практически важных задач может быть сформулировано в виде: среди всех возможных значений некоторо-

го набора  $\theta$  из множества  $\Theta$ , удовлетворяющих большому набору ограничений (условий)

$$\{\text{conditions}(\theta, x)\}_{x \in \mathbb{X}},$$

выбрать одно или несколько, которым удовлетворяют дополнительные требования к некоторой целевой функции  $f(\theta)$  (например, минимизация, или равенство нулю, или принадлежность какому-то заданному множеству). Обычно величины  $x \in \mathbb{X} \subset \mathbb{R}^d$  трактуются как поступающие на вход обучающейся системе сигналы (стимулы). Это могут быть текущие пространственно-временные координаты, исследуемого объекта, электромагнитные или акустические сигналы и т. п.

На практике множество  $\mathbb{X}$  далеко не всегда можно предполагать конечным и даже ограниченным. Типичная трудность задачи поиска (выбора) подходящего  $\theta$  заключается в размере множества  $\mathbb{X}$ . Если оно все же конечное, но достаточно большое, то это не существенно облегчает решение, так как алгоритмы, основанные на переборе возможных вариантов не смогут закончить свою работу за реальное время.

## 2.1. Метод Монте-Карло

Истоки теории рандомизированного обучения (Randomized Learning Theory) [1] лежат в идеях метода статистического моделирования Монте-Карло, предложенного в Лос-Аламосе Джоном фон Нейманом, Николасом Метрополисом и Станиславом Уламом в ходе работ над Манхэттенским проектом [2]. Первоначально стохастический подход использовали для аппроксимации многомерных интегралов в уравнениях переноса, возникших в связи с задачей о движении нейтрона в изотропной среде. В университетских курсах математики обычно приводят следующий пример способа вычисления интеграла

$$f(\theta) = \int_{x \in \mathbb{X}} F(\theta, x) dx,$$

которым удобно пользоваться при невозможности аналитического интегрирования. Пусть множество  $\mathbb{X}$  является параллелепипедом с объемом  $V(\mathbb{X})$  и предположим, что значения функции  $F$  ограничены:  $0 \leq F(\theta, x) \leq F_{\max}$ . Геометрический смысл интеграла — объем

под графиком функции  $F$ . Зафиксируем натуральное число  $N$ , выберем в параллелепипеде  $\Pi \times [0, F_{\max}] \subset \mathbb{R}^{d+1}$  случайно  $N$  точек  $z_1, z_2, \dots, z_N$  и подсчитаем среди них число  $K$  тех, у которых последняя координата не превосходит значения функции  $F$  в соответствующей точке, определяемой первыми  $d$  координатами. Хорошей оценкой для значения интеграла будет

$$f(\theta) \approx V(\mathbb{X})F_{\max} \frac{K}{N}.$$

Причем необходимую точность приближения можно гарантировать *априори* при выборе  $N$  в силу закона больших чисел и неравенства Чебышева

$$\forall \epsilon > 0 \quad \mathbb{P} \left\{ \left| \frac{f(\theta)}{V(\mathbb{X})F_{\max}} - \frac{K}{N} \right| \geq \epsilon \right\} \leq \frac{f(\theta)(V(\mathbb{X})F_{\max} - f(\theta))}{N\epsilon^2(V(\mathbb{X})F_{\max})^2}.$$

При небольшой размерности  $d$  множества  $\mathbb{X}$  во многих случаях можно при вычислении интегралов пользоваться и традиционными детерминированными приближенными методами, но при  $d \gg 1$  на практике фактически ничем нельзя воспользоваться кроме метода Монте-Карло, сложность которого не зависит  $d$ . Метод Монте-Карло может оказаться более предпочтительным и при небольших  $d$ , когда функция задана неявно, а необходимо определить область, заданную в виде сложных неравенств.

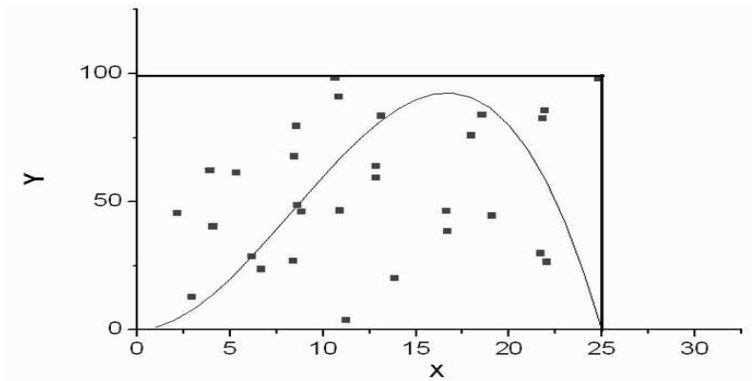


Рис. 1: Интегрирование по Монте-Карло.

На рис. 1 приведен пример интегрирования одномерной функции методом Монте-Карло. Для этого 30 случайно выбранных в

прямоугольнике  $[0; 25] \times [0; 100]$  точек были сравнены с 30-ю соответствующими значениями функции и было подсчитано количество  $K = 16$ , попавших под график. В результате получаем  $f \approx 25 \cdot 100 \cdot 14/30 = 1166\frac{2}{3}$ . При этом вероятность того, что относительная ошибка превысит 5% примерно равна 0,11. Конечно в одномерном случае ( $d = 1$ ) по 30 измерениям можно было бы хорошо аппроксимировать интеграл и методом разбиения  $\mathbb{X}$  на 30 отрезков, но при  $d \gg 1$ , разбивая  $\mathbb{X}$  на отрезки, ту же точность можно обеспечить при  $30^d$  измерений, что несоизмеримо велико с  $N = 30$  для метода Монте-Карло.

Изобретение и развитие метода Монте-Карло неслучайно совпало по времени с созданием первых вычислительных машин. Первые ЭВМ уже могли с большой скоростью генерировать псевдослучайные числа, что резко расширило круг задач, для решения которых стохастический подход оказался более эффективным, чем другие математические методы. В 1950-х годах метод активно использовался для расчетов при разработке водородной бомбы.

## 2.2. Сценарный подход

Многие практические задачи сводятся к решению проблемы робастной выпуклой оптимизации (Robast Convex Problems, RCP), в которых оптимизируются выпуклые функции и множество ограничений задается также с помощью выпуклых функций. В общем виде задачи робастной выпуклой оптимизации, как правило, NP-сложные, так как в них обычно для решения требуется проверка огромного количества условий-ограничений. Рандомизированный подход позволяет при априорном задании параметров уровня  $\epsilon$  и конфиденциальности  $\beta$  получать решение, которое с задаваемой вероятностью  $(1 - \beta)$  будет решением оптимизационной задачи, удовлетворяющим почти всем ограничениям, за исключением множества меры не большей  $\epsilon$ . В 2008 году статья Дж. Калафиори и М. Кампи [3] с описанием основных идей сценарного подхода была выделена IEEE CSS как лучшая статья 2006 года.

Пусть  $\theta \in \Theta \subset \mathbb{R}^n$  — параметр конструирования некоторой системы или объекта управления. Рассмотрим семейство выпуклых задач оптимизации, параметризованных  $x \in \mathbb{X} \subseteq \mathbb{R}^d$ :

$$RCP : \min_{\theta \in \Theta \subseteq \mathbb{R}^n} c^T \theta \text{ при ограничениях } \{F(\theta, x) \leq 0, x \in \mathbb{X}\},$$

где оптимизация функции  $f(\theta) = c^T \theta$  проводится по  $\theta \in \Theta$ , множество  $\Theta \subset \mathbb{R}^n$  выпукло и замкнуто,  $c \in \mathbb{R}^n$  — некоторый заданный вектор, а функция  $F : \Theta \times \mathbb{X} \rightarrow \mathbb{R}$  выпукла по  $\theta$  для всех  $x \in \mathbb{X}$ . В типичных случаях мощность множества  $\mathbb{X}$  может быть бесконечной, т. е. оно может содержать бесконечное число элементов  $x$ .

Заметим, что рассмотрение случая скалярной функции  $F$  никоим образом не снижает общности рассуждений, так как случай многих ограничений

$$\{F_i(\theta, x) \leq 0, x \in \mathbb{X}, i = 1, 2, \dots, I\},$$

легко к нему сводится через  $F(\theta, x) = \max_i F_i(\theta, x)$ .

Важным для приложений частным случаем *RCP* являются задачи робастного линейного программирования, в которых  $F(\theta, x)$  — аффинные по  $\theta$ , и полуопределенного программирования, в которых  $F(\theta, x) = \lambda_{\max}(A(\theta, x))$ , где  $\lambda_{\max}(A)$  — максимальное собственное число симметричной матрицы  $A$ , и

$$A(\theta, x) = A_0(x) + \sum_{i=1}^n \theta_i A_i(x), \quad A_i(x) = A_i^T(x).$$

*Конструирование для “худшего” случая*, с одной стороны, приводит к необходимости решения NP-сложных задач, а с другой — часто приносит в систему чрезмерный консерватизм, поскольку вынуждает фокусироваться на всех вырожденных случаях, вероятность появления которых на практике не велика.

В парадигме *вероятностного (рандомизированного) робастного конструирования* предполагается задание вероятностной меры на множестве неопределенностей  $\mathbb{X}$ . Тогда для задаваемого уровня вероятности  $\epsilon \in (0, 1)$  ищется параметр конструирования  $\theta$ , минимизирующий целевую функцию  $c^T \theta$  и удовлетворяющий почти всем ограничениям за исключением такой небольшой части из них, что ее мера (вероятность) не превосходит заранее заданного уровня  $\epsilon$ .

Пусть на  $\mathbb{X}$  задана  $\sigma$ -алгебра и на ней определена вероятность  $\mathbb{P}$ . Для всякого  $\theta \in \Theta$  *вероятностью нарушения* назовем величину

$$V(\theta) \doteq \mathbb{P}\{x \in \mathbb{X} : F(\theta, x) > 0\}$$

(здесь предполагается, что событие  $\{x \in \mathbb{X} : F(\theta, x) > 0\}$  является элементом  $\sigma$ -алгебры). Будем говорить, что решение  $\theta \in \Theta$  является *решением уровня  $\epsilon$* , если  $V(\theta) \leq \epsilon$ .

Пусть  $x_1, \dots, x_N$  —  $N$  независимых реализаций, полученных по одинаковой вероятностной мере  $P$ . Сценарная задача является дискретизацией задачи  $RCP$  и представляет собой решение следующей задачи выпуклого программирования

$$RCP_N : \min_{\theta \in \Theta \subseteq \mathbb{R}^n} c^T \theta, \text{ при } \{F(\theta, x_i) \leq 0, x_i \in \mathbb{X}, i = 1, 2, \dots, N\}.$$

В [3] доказан следующий интересный математический результат. Фиксируем вещественные параметр уровня  $\epsilon \in (0, 1)$  и параметр конфиденциальности  $\beta \in [0, 1]$  и положим

$$N \geq N(\epsilon, \beta) = \left\lceil \frac{2}{\epsilon} \ln \frac{2}{\beta} + 2n + \frac{2n}{\epsilon} \ln \frac{2}{\epsilon} \right\rceil.$$

(Здесь и далее  $[\cdot]$  — функция целой части числа). Тогда с вероятностью не меньшей, чем  $1 - \beta$  либо сценарная задача  $RCP_N$  неразрешима и тогда первоначальная задача робастной выпуклой оптимизации  $RCP$  также неразрешима, либо  $RCP_N$  разрешима и тогда ее оптимальное решение является решением уровня  $\epsilon$  для первоначальной задачи  $RCP$ .

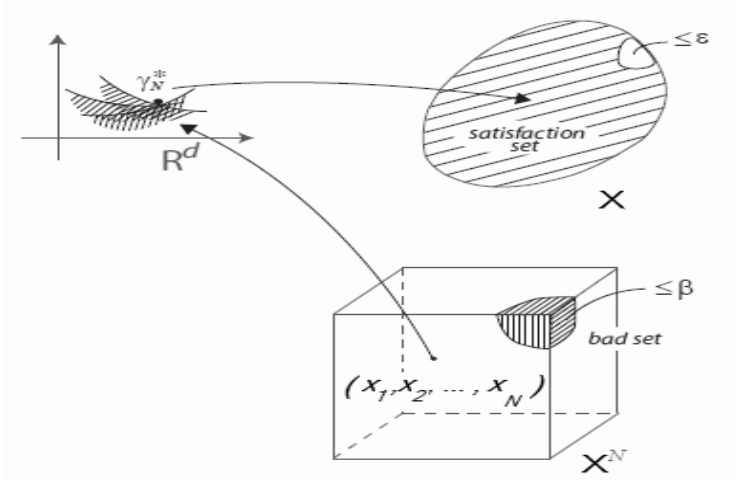


Рис. 2: Сценарный подход [3].

Рис. 2 иллюстрирует сценарный подход. С вероятностью не меньшей  $1 - \beta$  мы выбираем точки  $x_1, \dots, x_N$  вне “bad set”. Получающе-

еся оптимальное решение удовлетворяет почти всем неопределенностям, кроме множества меры  $\epsilon$ .

Кроме описанных в [3] приложений в задачах теории управления сценарный подход может быть использован и при обнаружении неисправностей, “разладки” или нетрадиционного поведения в самых разнообразных системах. В [4] описана идея рандомизированного подхода к обнаружению разрывов функции, инспирированная сценарным подходом.

### 2.3. Распознавание образов

На протяжении последних десятилетий в связи со стремительным развитием цифровых технологий наблюдается значительный рост объемов хранимых и перерабатываемых данных. Однако увеличение количества информации не означает непосредственного увеличения объемов знаний. В такой ситуации все более востребованными становятся новые математические методы, которые позволяли бы распознавать образы, структурировать информацию и находить объективные закономерности в больших объемах данных. В теории обучения типичной является задача классификация входных сигналов (стимулов, объектов). В процессе классификации обнаруживаются признаки, которые характеризуют группы объектов исследуемого набора данных – классы (кластеры). По этим признакам каждый сигнал можно отнести к тому или иному классу. Результатом кластеризации является разбиение сигналов на группы в условиях, когда классы заранее не определены. Получающееся разбиение естественным образом характеризует структуру множества данных и может быть использовано в дальнейшем для ее определения.

С содержательной точки зрения достаточно часто процесс распознавания образов трактуется как классификация входных сигналов  $x$  из некоторого множества  $\mathbb{X}$ , заключающаяся в построении правила сопоставления каждой точке  $x \in \mathbb{X}$  некоторого образа (класса)  $\mathbb{X}_k$ . Выбор правила классификации порождает разбиение множества  $\mathbb{X}$  на классы. Будем считать, что правило классификации однозначно определяется конечномерным набором  $\eta$  и задана функция  $l(\eta)$  возвращающая количество классов при классификации по правилу  $\eta$ . Всякий способ классификации  $\eta$  связан с потерями, которые обычно характеризуются с помощью штрафных функ-



ций стоимости  $q_k(\eta, x)$  при отнесении точки  $x$  к классу с номером  $k$ . В типичных случаях, когда  $\mathbb{X}$  — вещественное векторное пространство, значения штрафных функций  $q_k(\eta, x)$  возрастают при удалении  $x$  от центра соответствующего образа (класса).

Рассмотрим следующее правило классификации: при заданных  $\eta$  и функциях  $q_k(\eta, x)$ ,  $k = 1, \dots, l(\eta)$ , входной сигнал  $x$  из множества  $\mathbb{X}$  относится к тому классу  $\mathbb{X}_k$  с наименьшим номером  $k = k(x)$ , для которого значение соответствующей штрафной функции  $q_k(\eta, x)$  минимально

$$k(x) = \min\{\arg \min_{i \in 1:l(\eta)} q_i(\eta, x)\}.$$

Например, если на  $\mathbb{X}$  задана норма  $\|\cdot\|$  и в множестве данных  $l$  классов, а  $\eta^l$  — набор векторов центров классов:  $\eta^l = (\theta^1, \theta^2, \dots, \theta^l)$ , то можно считать  $\eta = \{l, \eta^l\}$  и в качестве  $q_k(\eta, x)$  можно задать расстояние до центра  $k$ -го класса  $\theta_k$ :  $q_k(\eta, x) = \|\theta_k - x\|^2$ . При этом множество  $\mathbb{X}$  разбивается на  $l$  классов  $\mathbb{X}_1(\eta), \mathbb{X}_2(\eta), \dots, \mathbb{X}_l(\eta)$  таким образом, что к классу (подмножеству)  $\mathbb{X}_k(\eta)$  относятся все точки  $x$ , находящиеся к центру  $\theta_k$  ближе, чем к любому другому. Интеграл  $\int_{\mathbb{X}_k(\eta)} \|\theta_k - x\|^2$  определяет рассеяние точек  $x$  в подмножестве  $\mathbb{X}_k(\eta)$ .

Предположим, что на множестве  $\mathbb{X}$  задано распределение  $P(\cdot)$ . Определим функционал качества кластеризации по правилу  $\eta$ :

$$F(\eta) = \sum_{k=1}^{l(\eta)} \int_{\mathbb{X}_k(\eta)} q_k(\eta, x) P(dx).$$

Тогда задача кластеризации множества данных из  $l$  классов состоит в определении набора центров  $\eta_\star^l$ , минимизирующего суммарную стоимость разбиения. Задача устойчивой кластеризации является обобщением для случая нахождения заранее неизвестного оптимального значения количества классов  $l_\star$  и соответствующего набора центров  $\eta_\star^{l_\star}$ .

Если в каком-то смысле можно было бы говорить о дифференцируемости функционала  $F$ , то искомый набор центров  $\eta_\star$  должен был бы удовлетворять уравнению  $\nabla F(\eta_\star) = 0$ , которое можно было бы попытаться решить традиционными средствами. Сложность рассматриваемой задачи заключается в недифференцируемости функционала  $F$ . При известном распределении  $P(\cdot)$  сформулированные задачи все-таки могут быть точно или приближенно решены. Нетрудно убедиться, что в описанном выше примере

множества  $\mathbb{X}_k(\eta)$  имеют вид многогранников, а минимизирующий функционал среднего риска набор центров  $\eta_{l^*}^l = (\theta_1^*, \theta_2^*, \dots, \theta_{l^*}^*)$  совпадает с *центрами тяжести* этих множеств, т. е.

$$\theta_k^* = \frac{\int_{\mathbb{X}_k(\eta_*)} xP(dx)}{\int_{\mathbb{X}_k(\eta_*)} P(dx)}, \quad k = 1, 2, \dots, l^*.$$

Приведенные соображения отвечают интуитивному представлению о разбиении множества  $\mathbb{X}$  на  $l_*$  непересекающихся классов, причем центры тяжести соседних множеств находятся на прямой, ортогональной разделяющей множества грани.

В системах, работающих в режиме реального времени в условиях изменяющейся со временем обстановки, распределение  $P(\cdot)$  очень часто бывает неизвестно. Будем предполагать, что в режиме реального времени на вход поступает последовательность  $x_1, x_2, \dots$  сигналов, порожденная неизвестным распределением  $P(\cdot)$ . *Требуется* предложить алгоритм построения последовательности оценок  $\{\hat{\eta}_i\}$  набора  $\eta_*$ , минимизирующего определенный выше функционал среднего риска. Решение задачи дополнительно осложняется тем, что на практике функции  $q^k(\cdot, \cdot)$ ,  $k = 1, 2, \dots, l(\eta)$  не всегда заданы аналитически, но доступны измерению их значения (может быть с помехами):

$$y^k(\eta, x) = q_k(\eta, x) + v_k, \quad k = 1, 2, \dots, l(\eta).$$

В 2003 году в монографии [5] был предложен простой с вычислительной точки зрения рекуррентный рандомизированный алгоритм для автоматического разбиения множества данных на заданное заранее количество классов, теоретическое обоснование которого было сделано позже в совместной работе с О.А. Измаковой [6]. За последние пять лет под руководством О.Н. Граничина по теме “Рандомизированные алгоритмы при распознавании образов” были защищены две кандидатские диссертации О.А. Измаковой (2005, по специальности 01.01.09) и Д.С. Шалымовым (2009, по специальности 05.13.11). В работах Д.С. Шалымова основное внимание уделялось вопросам устойчивости кластеризации. В итоге удалось совместно с коллегами из Израиля предложить, обосновать и апробировать принципиально новый эффективный рандомизированный метод определения достоверного интервала для действительного значения кластеров в множестве данных, базирующийся на описанном выше сценарном подходе [7].

Основная идея нового рандомизированного алгоритма устойчивой кластеризации состоит в построении по небольшому количеству точек нескольких равномерных аппроксимаций индексной функции специального вида, которая имеет “скачок” в точке, соответствующей истинному количеству кластеров. В традиционных предшествовавших алгоритмах необходимо было вычислять соответствующую индексную функцию для всех возможных значений, что на практике при большой мощности исследуемого множества труднореализуемо. В докладе [8], были приведены примеры возможного использования сценарного подхода для определения количества кластеров в множествах данных очень большого объема. Исследованный рандомизированный алгоритм поиска наиболее значительного скачка индексной функции типа предложенной в [9] во многом базируется на предшествующей работе [10], в которой предложен новый оригинальный алгоритм поиска разрыва у кусочно-непрерывной функции. Имитационные эксперименты показывают работоспособность нового рандомизированного алгоритма устойчивой кластеризации при определении нескольких тысяч кластеров в условиях неопределенности на порядок превышающей их действительное количество. При этом использование нового алгоритма позволяет существенно сократить общую вычислительную сложность процесса поиска кластеров.

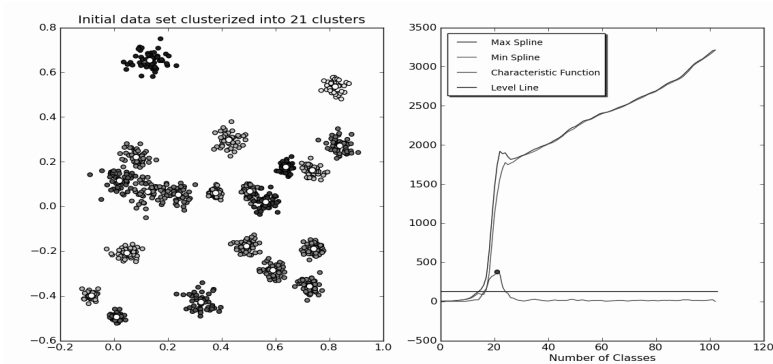


Рис. 3: Определение количества кластеров в множестве данных.

На рис. 3 показаны результаты работы рандомизированного алгоритма по поиску количества кластеров в множестве данных, по-

лученные в дипломной работе М.А. Морозкова (2010). Искомое количество кластеров 21 из 100 возможных было определено по результатам трех аппроксимаций индексной функции по пяти точкам (всего 15 измерений). Более существенный выигрыш показывает применение нового алгоритма для исследования данных с несколькими тысячами кластеров, при которых удается использовать всего несколько десятков вычислений индексной функции.

### 3. Минимизация функционалов типа среднего риска

Как уже отмечалось, проблема оптимизации того или иного функционала встает во многих практических приложениях. Иногда экстремальные значения можно найти аналитически, однако зачастую инженеры имеют дело с неизвестным функционалом, значение которого можно только вычислять в задаваемых точках. При этом возможно появление неконтролируемых неопределенностей различной природы, как статистических, так и нерегулярных (например, неизвестных, но ограниченных, для которых традиционные предположения о статистической природе, независимости и центрированности не выполнены).

Для неопределенностей статистической природы в современных практически значимых задачах финансовой математики, теории игр, распознавания биологических видов, в Интернет-системах, при планировании электросетей, предотвращении эпидемий и лечении инфицированных все чаще обоснованно предполагают степенное распределение, при котором может отсутствовать второй статистический момент.

Пусть  $(\Omega, \mathcal{F}, P)$  — вероятностное пространство, и  $F(\theta, x) : \mathbb{R}^n \times \mathbb{X} \rightarrow \mathbb{R}$  — дифференцируемая по первому аргументу функция,  $\mathbb{X} \subset \mathbb{R}^d$ . Будем считать, что в каждый момент времени  $i = 1, 2, \dots$  в выбираемых экспериментатором точках измерения  $\theta_1, \theta_2, \dots$  (план наблюдения) доступно наблюдению с аддитивной помехой  $v_i$  значение функции

$$y_i = F(\theta_i, x_i) + v_i, \quad (1)$$

где  $x_i \in \mathbb{X}$  — неконтролируемая последовательность независимых случайных векторов, имеющих одинаковое, вообще говоря, неизвестное распределение  $P_x(\cdot)$ .

Рассмотрим задачу о построении по наблюдениям  $y_1, y_2, \dots$  последовательности оценок  $\{\widehat{\theta}_i\}$  неизвестного вектора  $\theta^*$ , минимизирующей функцию  $f(\theta)$  типа функционала среднего риска:

$$f(\theta) = E\{F(\theta, x)|\theta\} \rightarrow \min. \quad (2)$$

(Здесь и далее  $E, E\{\cdot\}, E\{\cdot|\cdot\}$  — символы математического ожидания (МО) и условного МО).

Обычно исследуется случай  $F(\theta, x) = f(\theta)$ . Сделанное обобщение позволяет, например, учесть случай мультипликативной неопределенности:  $F(\theta, x) = xf(\theta)$  и  $Ex = 1$ .

В зависимости от конкретных условий задачи, свойств функции  $F$  и неопределенностей  $x_i$  и  $v_i$  сходимость последовательности оценок понимается в смысле одного из следующих определений.

1. Последовательность оценок  $\{\widehat{\theta}_i\}$  величины  $\theta$  называется состоятельной, если  $\forall \epsilon > 0 \lim_{i \rightarrow \infty} P\{\|\widehat{\theta}_i - \theta^*\| > \epsilon\} = 0$ .

2. Последовательность оценок  $\{\widehat{\theta}_i\}$  величины  $\theta^*$  называется сильно состоятельной, если сходимость оценок к  $\theta^*$  выполняется с вероятностью единица  $\widehat{\theta}_i \rightarrow \theta^*$  при  $i \rightarrow \infty$ .

Наряду с (2) часто рассматривается и нестационарная постановка задачи. Пусть задано семейство функций  $\{F_{\xi}(\theta, x)\}_{\xi \in \Xi}$ , дифференцируемых по первому аргументу,  $\theta_1, \theta_2, \dots$  — выбираемая экспериментатором последовательность точек измерения (план наблюдения), в которых в каждый момент времени  $i = 1, 2, \dots$  доступно наблюдению с аддитивной помехой  $v_i$  значение функции  $F_{\xi_i}(\theta_i, x_i)$

$$y_i = F_{\xi_i}(\theta_i, x_i) + v_i, \quad (3)$$

где  $\xi_i$  — неконтролируемая последовательность:  $\xi_i \in \Xi$ . *Требуется* по наблюдениям  $y_1, y_2, \dots$  построить последовательность оценок  $\{\widehat{\theta}_i\}$  неизвестных векторов  $\{\theta_i\}$ , минимизирующих функции  $f_i(x)$  типа функционала среднего риска:

$$f_i(x) = E\{F_{\xi_i}(\theta, x)|\theta\} \rightarrow \min. \quad (4)$$

Задача (2) является частным случаем (4) при  $\theta_i \equiv \theta$  и  $\Xi = \{\xi_1\}$ .

Для нестационарной задачи оптимизации (4) естественна постановка задачи о стабилизации последовательности оценок и определении границу стабилизации  $L > 0$  (стабилизируется) по отношению к последовательности  $\{\theta_i\} \subset \mathbb{R}^n$  в следующем смысле:

$$\overline{\lim}_{i \rightarrow \infty} (E\|\widehat{\theta}_i - \theta_i\|_{\rho}^{1/\rho}) \leq L.$$

(Здесь и далее  $\|a - b\|_\rho = \left( \sum_{j=1}^n |a^{(j)} - b^{(j)}|^\rho \right)^{\frac{1}{\rho}}$ ,  $\forall a, b \in \mathbb{R}^n$  с компонентами  $a^{(j)}, b^{(j)}, j = 1, \dots, n$  и  $\rho \in (1, 2]$ ).

### 3.1. Рандомизированные алгоритмы стохастической оптимизации

Комбинация более ранних идей обобщенных стохастических градиентов, псевдоградиентных методов и операторов усреднения привела в конце 1980-х к появлению рандомизированных алгоритмов стохастической аппроксимации (РАСА), называемых в англоязычной литературе Simultaneous Perturbation Stochastic Approximation (SPSA), в работах О.Н. Граничина [11], Дж. Спалла [12], Б.Т. Поляка и А.Б. Цыбакова [13], и Х.-Ф. Чена, Т. Дункан и Б. Пасик-Дункан [14]. Они основаны на аппроксимации градиента вдоль случайно выбранного направления — рандомизации, которая опирается на некоторые контролируемые случайные величины, влияющие на результаты наблюдений, существующие в системе или добавляемые экспериментатором. Обычно рандомизация процесса наблюдений позволяет повысить скорость работы алгоритма оптимизации в случае большой размерности  $n$  и его устойчивость к помехам. Обозначим пробное возмущение  $\Delta_i \in \mathbb{R}^n$ ,  $i = 1, 2, \dots$  и  $P_{\Delta_i}(\cdot)$  — его распределение.

Пусть  $\hat{\theta}_0 \in \mathbb{R}^n$  и задана последовательность  $\{\Theta_i\}$  выпуклых множеств  $\Theta_i \subset \mathbb{R}^n$ , содержащих начиная с некоторого  $N > 0$  точку/и минимума/-ов функционала (2)/-ов (4). Например,  $\{\Theta_i\}$  — последовательность расширяющихся до бесконечности шаров. Рассмотрим свойства оценок следующих трех алгоритмов типа случайного поиска:

$$\bar{\theta}_i = \hat{\theta}_{i-1} + \beta_i \Delta_i, y_i = F(\bar{\theta}_i, x_i) + v_i,$$

$$\hat{\theta}_i = \mathcal{P}_i(\hat{\theta}_{i-1} - \frac{\alpha_i}{\beta_i} \mathcal{K}_i(\Delta_i) y_i), \quad (5)$$

с одним и двумя измерениями на итерации

$$\bar{\theta}_{2i} = \hat{\theta}_{i-1} + \beta_i \Delta_i, \bar{\theta}_{2i-1} = \hat{\theta}_{i-1} - \beta_i \Delta_i,$$

$$\hat{\theta}_i = \mathcal{P}_i(\hat{\theta}_{i-1} - \frac{\alpha_i}{2\beta_i} \mathcal{K}_i(\Delta_i)(y_{2i} - y_{2i-1})), \quad (6)$$

$$\bar{\theta}_{2i} = \hat{\theta}_{i-1} + \beta_i \Delta_i, \quad \bar{\theta}_{2i-1} = \hat{\theta}_{i-1},$$

$$\hat{\theta}_i = \mathcal{P}_i(\hat{\theta}_{i-1} - \frac{\alpha_i}{\beta_i} \mathcal{K}_i(\Delta_i)(y_{2i} - y_{2i-1})), \quad (7)$$

где  $\mathcal{P}_i$  — проекторы на  $\Theta_i$ , причем для алгоритма (5)  $\Theta_i$  — компакты, а  $\mathcal{K}_i(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  — некоторые вектор-функции (ядра), например,  $\mathcal{K}_n(\Delta) = \Delta$ .

О.Н. Граничиным установлена состоятельность и сильная состоятельность методов с пробным возмущением (алгоритмов РА-СА) при условии почти произвольных помех [5, 11]. Б.Т. Поляк и А.Б. Цыбаков обосновали минимаксную асимптотическую оптимальность скорости сходимости оценок алгоритмов (5) и (6) в том смысле, что порядок ее не может быть улучшен никаким другим итеративным алгоритмом [13]. Дж. Спалл показал асимптотическое сокращение количества измерений для методов с пробным возмущением типа (6), по сравнению с процедурой Кифера-Вольфовица, что является существенным при больших  $n$  [12].

В задаче (4) алгоритмы стохастической аппроксимации с пробным возмущением (5)-(7) обычно используются с постоянными шагами  $\alpha_n \equiv \alpha$ ,  $\beta_n \equiv \beta$ . На рис. 4 показана блок-схема алгоритма (7) рекуррентной оптимизации нестационарного функционала.

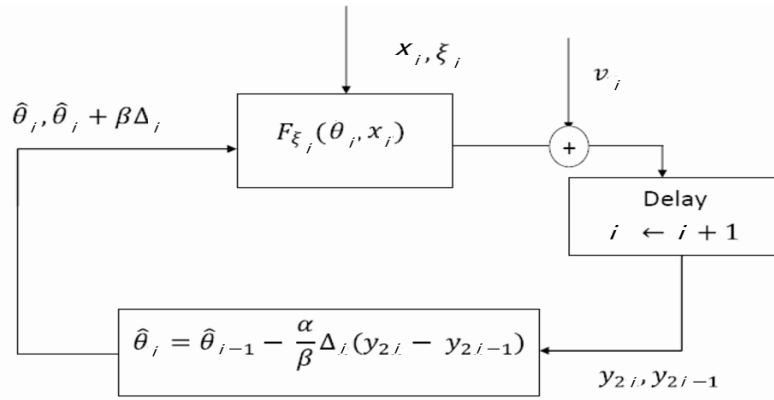


Рис. 4: Блок-схема алгоритма (7) в задаче нестационарной оптимизации.

При исследовании свойств получающихся оценок обычно пред-

полагают выполненными ряд условий на функции  $F$  и неопределенности  $x_i$  и  $v_i$ :

- (А) Сильная выпуклость  $f(\theta)$  в точке минимума.
- (В) Ограниченность скорости изменения градиента  $\nabla F(\cdot, w)$ .
- (С) Условие перестановки операторов интегрирования и дифференцирования для  $F(\theta, x)$  и  $\nabla F(\theta, x)$ .
- (D) Ограниченность статистических моментов  $F(\theta, x)$  и  $\|\nabla F(\theta, x)\|_\rho$  в точке минимума.
- (Е) Ограниченность колебаний функции  $F$  в последовательные моменты времени.
- (F) Ограниченность дрейфа точек минимума.
- (G) Неизвестные помехи наблюдения  $v_i$  либо детерминированные и ограниченные, либо случайные независимые с  $\{\Delta_i\}$ , имеющие ограниченный момент степени  $\rho$ .
- (H) Будем считать пробное рандомизированное возмущение последовательностью независимых случайных величин с независимыми между собой компонентами.
- (I) Условия на размеры шагов алгоритмов  $\alpha_n, \beta_n \geq 0$  стандартные для алгоритмов типа стохастической аппроксимации.

Исследованию свойств оценок рандомизированных алгоритмов стохастической аппроксимации (РАСА) была посвящена существенная часть докторской диссертации О.Н. Граничина, основные результаты которой были опубликованы в [5]. В 2006-10 гг. на кафедре системного программирования под руководством О.Н. Граничина по свойствам оценок РАСА были защищены две кандидатские диссертации С.С. Сыроевым (2006, по специальности 05.13.11) и А.Т. Вахитовым (2009, по специальности 01.01.09), основные результаты которых опубликованы в [15] и [16]:

*Если выполнены условия (А) для  $f(\theta)$ , (В)-(Е) для  $F(\theta, x)$ , а также (G)-(I),*

**тогда оценки алгоритмов (5)-(7) являются состоятельными и  $E\|\hat{\theta}_i - \theta^*\|_\rho^2 \rightarrow 0$  при  $i \rightarrow \infty$ . При некоторых дополнительных условиях можно получить сильную состоятельность оценок или характеристики скорости сходимости.**

Для нестационарного случая при постоянных размерах шагов:  $\alpha_i \equiv \alpha$ ,  $\beta_i \equiv \beta$ , если равномерно по  $i$  выполнены условия (А) для  $f_i(\theta)$ , (В)-(Е) для  $F_{\xi_i}(\theta, x_i)$ , (F) для дрейфа минимума, (G) для аддитивных помех наблюдения, (H) для рандомизированного возмущения,



то можно специальным образом так определить некоторые постоянные  $K$  и  $L$ , что выполняется

$$E\|\hat{\theta}_i - \theta_{mi}\|_\rho^\rho \leq K^i E\|\hat{\theta}_0 - \theta_0\|_\rho^\rho + (1 - K^i) \frac{L}{1 - K},$$

и последовательность оценок  $\{\hat{\theta}_i\}$  имеет границу стабилизации, равную  $\bar{L} = \frac{L}{1-K}$  ( $m = 1$  или  $2$ ).

### 3.2. Оптимизация работы сервера, обрабатывающего очередь заданий

Для апробирования методов адаптивной оптимизации в задаче повышения эффективности обслуживания одним сервером очереди заданий в дипломных работах Я.В. Волкович (2005) и Ле Ч.К. (2007) была разработана модель, имитирующая работу сервера, и были проведены исследования применимости рандомизированных алгоритмов стохастической оптимизации при настройке наилучшего значения параметра диспетчеризации сервера в зависимости от реальных входных данных.

Проблема эффективного обслуживания сервером очереди заданий — одна из классических задач теории массового обслуживания. Существует множество методов для ее решения, были разработаны различные алгоритмы и проведены многочисленные исследования. Возможности использования адаптивных методов в этой задаче не столь детально изучены. В частности, ряд способов предлагается и исследуется в книге [17] и в статье [18]. Их обоснование базируется на использовании техники “анализа бесконечно малых возмущений”. В [19] было предложено использовать для адаптивной подстройки изменяемого параметра сервера рандомизированный алгоритм стохастической аппроксимации.

Рассмотрим для примера загруженность обычной корпоративной сети. В начале и конце рабочего дня в ней преобладают кратковременные и частые запросы к серверу, а в середине дня задания поступают реже и становятся более трудоемкими. Такие колебания могут возникать не только в течении суток, но и зависеть от сезона, дня недели и т. п. Заметим, что для более результативной работы сервера можно подстраивать его характеристики под тип поступающих заданий, но как определить границы периодов? Возможно провести некоторое исследование-мониторинг и получить приближительные искомые значения. Этот способ не очень хорош, потому

как в реальной ситуации выявить прямую зависимость от факторов сложно, и, кроме того, подобное исследование весьма дорого и требует повторных работ даже при незначительных изменениях. Хотелось бы иметь возможность реагировать на изменения этих значений в режиме реального времени в зависимости от вида поступающих заданий.

Формализуем постановку задачи о повышении эффективности сервера. Пусть он используется для обслуживания очереди заданий, процесс поступления которых является случайным. Будем считать, что вероятностное распределение времени обслуживания задания сервером зависит от вещественного параметра  $\theta$ , который требуется выбрать с целью минимизации среднего времени ожидания клиентами  $L(x)$  вместе с некоторой стоимостью использования параметра  $x$

$$f(\theta) = q(\theta) + L(\theta) \equiv q(\theta) + \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N E y_i(\theta) \rightarrow \min,$$

где  $y_i(\theta)$  — время, которое задание, законченное  $i$ -м по счету, ожидало (“простаивало”) в сервере до момента своего завершения. *Требуется* найти параметр  $\theta^*$ , минимизирующий  $f(\theta)$  по  $\theta$  из некоторого компактного множества  $\Theta \subset \mathbb{R}$  (области определения).

Вообще говоря, функцию  $L(\theta)$  очень трудно вычислить. Пусть  $\theta$  фиксировано. Можно, наблюдая за очередью длительное время (фиксируя время поступления заказа, время обслуживания и время отправки для каждого клиента), использовать полученные данные для оценки  $\hat{G}$  производной функционала качества  $\nabla f(\theta)$  для текущего значения  $\hat{\theta}$ , получающегося в результате подсчета по некоторому градиентному алгоритму. Здесь термин оценка производной понимается в широком смысле, так как она может получаться смещенной. Фактически, мы обязаны предполагать, что получающаяся оценка производной зависит не только от выбранной точки  $\hat{\theta}$ , но и от некоторого набора  $x$  неконтролируемых возмущений (помех)

$$\hat{G} = G(\hat{\theta}, x),$$

определяющихся конкретными условиями функционирования сервера на выбранном для исследований интервале времени, включая фактически реализовавшуюся последовательность запросов. Основная трудность, общая для многих систем реального времени, заклю-

чается в том, что оптимизация должна быть получена на основании наблюдений только на конкретной траектории реализации случайной величины. У экспериментатора нет возможности выбирать значения функции в некоторой окрестности точки  $x$  для формирования оценки производной в этой точке, т. е. эту задачу оптимизации не решить традиционными средствами, стандартные конечно-разностные методы оценки производной функционала качества в рассматриваемой задаче малоприспособлены. Более адекватными подходами к решению являются алгоритмы основанные на использовании подсчитываемых на определенных интервалах времени значений эмпирических функционалов качества  $F(x, w)$ , которые при определенных естественных предположениях о регулярности входного потока и эксплуатационных характеристиках сервера являются несмещенными оценками значений функционала качества

$$f(\theta) = EF(\theta, x).$$

Кроме перечисленных трудностей с течением времени оптимальное значение  $\theta$  обычно дрейфует, например, как при колебаниях параметров входных потоков в корпоративных вычислительных сетях. Поэтому наряду с задачей оптимизации не менее актуальной является проблема о трекинге (отслеживании изменяющейся точки минимума). Более точно, пусть  $F_t(\theta, x)$  — случайная функция дискретного времени  $t = 1, 2, \dots$ , векторного параметра  $\theta$  и вектора неконтролируемых возмущений  $x$ . Обозначим через  $f_t$  “текущий” функционал среднего риска:

$$f_t(\theta) = E\{F_t(\theta, x)|\theta\}.$$

Точку минимума функционала  $f_t$  обозначим

$$\theta_t = \arg \min_{\theta} f_t(\theta).$$

Как и ранее будем считать, что, выбрав  $\theta$  на определенных интервалах времени, мы можем измерить значения соответствующего “текущего” эмпирического функционала  $F_t(\theta, x)$ . При этом мы естественно предполагаем, что скорость “дрейфа” оптимального параметра настолько низкая, что мы в состоянии получить целую серию значений  $F_t(\theta_i, x_i)$ ,  $i = 1, 2, \dots$

*Требуется* по наблюдениям (может быть с дополнительными помехами) за случайными величинами  $F_t(\theta_i, x_i)$ ,  $n = 1, 2, \dots$  по-

строить последовательность оценок  $\{\hat{\theta}_t\}$ , отслеживающих изменения параметра  $\theta_t$ , для которой  $\|\hat{\theta}_t - \theta_t\| \rightarrow \min$  в каком-нибудь смысле.

Будем предполагать, что процесс поступления этих заданий от клиентов носит случайный характер. Входные данные определяют последовательность пар чисел  $\{(t_i^{in}, d_i)\}$ , в которых первое значение соответствует времени прихода задания на сервер, а второе — времени, необходимому для его исполнения. Естественно, что для сервера значения  $d_i^{in}$  считаются неизвестными.

Для простоты считаем, что сервер обрабатывает задания следующим образом:

- задается шаг диспетчеризации  $\theta$  ( $\theta \in \Theta = [a; b]$ );
- первое поступившее задание начинает выполняться сервером сразу после поступления;
- поступающие задания попадают в естественную очередь;
- в каждый момент времени сервер обрабатывает только одно текущее задание;
- в моменты времени кратные выбранному шагу диспетчеризации  $\theta$ , если сервер свободен, то он переключается на выполнение первого в очереди задания, если он занят, то выполняющееся задание прерывается и отправляется в конец очереди, а сервер переключается на первое задание из очереди. На эту операцию “загрузки-выгрузки” сервер всегда затрачивает фиксированное время  $d_{load}$ .

Выберем натуральное достаточно большое число  $N$  (например,  $N = 1000$ ). Разобьем общее время работы сервера на определенные интервалы — такты:  $t_0, t_1, \dots, t_k, \dots$ , по правилу:

- $t_0$  — время начала работы сервера;
- $t_1$  — время окончания выполнения первых  $N$  заданий;
- ...
- $t_k$  — время окончания выполнения  $k$ -ой серии из  $N$  заданий;
- ...

Отметим, что вообще-то задавать такты можно различными способами: либо начиная новый интервал по завершении обработки группы из некоторого фиксированного числа заказов, либо после прихода на сервер фиксированного числа заданий, либо выбирая фиксированный интервал времени сервера, и т. п. Все разумные правила выбора тактов во многом эквивалентны.

Опишем способ получения значений  $y_k$  для эмпирического функционала  $F(\theta, x)$ . Считаем, что для каждого  $k$ -го такта каким-то

образом задано соответствующее значение оценки  $\hat{\theta}_k$  для регулируемого параметра  $\theta$  (интервала времени обслуживания задания на сервере). Для удобства перенумеруем задания, выполненные сервером, в порядке окончания их обработки. Каждому заданию наряду с парой чисел  $(t_i^{in}, d_i)$ , определенных уже в момент его поступления, сопоставим еще одно —  $t_i^{out}$  — время окончания его обработки. Определим значения эмпирической функции качества как среднее время бесполезного ожидания заданий до их завершения с учетом стоимости использования загрузки-выгрузки обработчика

$$y_k = \frac{(t_k - t_{k-1}) \cdot d_{load}}{\hat{\theta}_k} + \frac{1}{N} \sum_{t_i^{out} \in [t_{k-1}; t_k]} (t_i^{out} - t_i^{in} - d_i).$$

### Алгоритмы адаптации.

Выберем некоторые достаточно малые коэффициенты  $\alpha > 0$  и  $\beta > 0$  и в качестве пробного одновременного возмущения возьмем последовательность наблюдаемых одинаково распределенных независимых случайных чисел  $\{\Delta_t\}$ , равных  $\pm 1$  с одинаковой вероятностью. Обозначим  $\mathcal{P}_{[a;b]}(\cdot)$  — проектор в интервал  $[a; b]$ .

*Алгоритм SPСА с одним измерением:*

1. Положим  $k = 0$  и выберем некоторое начальное значение оценки  $\hat{\theta}_0$ .
2. В начале каждого  $k$ -го такта вычисляем  $\theta'_k = \mathcal{P}_{[a;b]}(\hat{\theta}_k + \beta \Delta_k)$ .
3. Запускаем сервер с значением параметра  $\theta = \theta'_k$ .
4. После завершения  $k$ -го такта подсчитаем новую оценку по правилу

$$\hat{\theta}_{k+1} = \mathcal{P}_{[a;b]}(\hat{\theta}_k - \frac{\alpha}{\beta} \Delta_k y_k).$$

5. Увеличиваем номер такта  $k = k + 1$ .
6. Переход к п. 2 (повтор действий заново).

*Алгоритм SPСА с двумя измерениями* отличается от предыдущего алгоритма только в двух операциях:

2. В начале каждого такта  $k$  вычисляем значение  $\theta'_k$ , но в этот раз оно зависит от четности счетчика: если  $k$  четное, то  $\theta'_k = \hat{\theta}_k$ , иначе  $\theta'_k = \mathcal{P}_{[a;b]}(\hat{\theta}_k + \beta \Delta_k)$ .
4. После завершения  $k$ -го такта подсчитаем новую оценку по правилу: если  $k$  четное, то  $\hat{\theta}_{k+1} = \hat{\theta}_k$ , иначе

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \frac{\alpha}{\beta} \Delta_k (y_k - y_{k-1}).$$

### Имитационное моделирование

Для имитации работы описанного выше сервера была написана программа, в которой задавать очередь входных заданий можно различными способами. Кроме того, интерфейс программы позволял запускать сервер (с новыми входными данными и нулевой очередью сервера), задавать паузу в работе сервера (все данные сохраняются), возобновлять работу сервера с сохраненными данными, завершать работу программы, задавать параметры  $\theta$  — интервал непрерывного времени обслуживания заданий сервером, и  $d_{load}$  — время, необходимое на загрузку–выгрузку заданий.

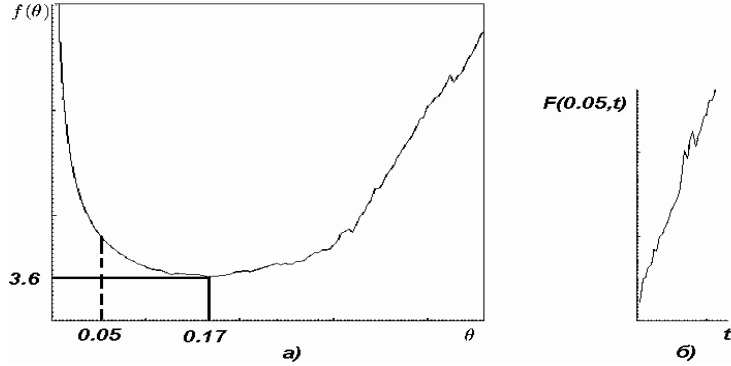


Рис. 5: Оптимизация функционала качества.

Входные данные представляют собой последовательность пар  $(t_i^{in}, d_i)$ . В программе предусмотрена возможность выбора входных данных или из файла, или в результате генерации псевдослучайной последовательности, задаваемой с помощью экспоненциальных распределений с параметрами  $\eta_{in}$  и  $\eta_d$  соответственно:

$$t_i^{in} = t_{i-1}^{in} - \frac{1}{\eta_{in}} \ln \omega_i^{in},$$

$$d_i = -\frac{1}{\eta_d} \ln \omega_i^d,$$

где  $\omega_i^{in}, \omega_i^d$  — некоторые реализации равномерно распределенной на  $[0; 1]$  случайной величины.

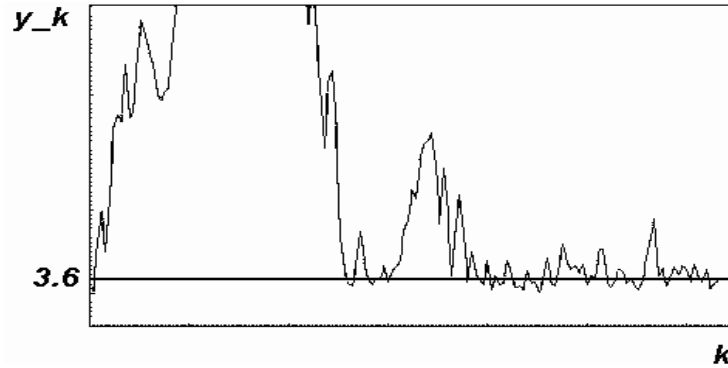


Рис. 6: Адаптация к оптимальному параметру.

Возможности имитационной модели позволяют, выбрав параметры входного потока, исследовать зависимость поведения значений эмпирического функционала качества от изменения параметра сервера  $\theta$  (шага диспетчеризации). На рис. 5а показан график зависимости от  $\theta$  усредненных значений эмпирического функционала при выборе  $d_{load} = 0.05$ ,  $\eta_{in} = 0.5$  и  $\eta_d = 0.75$ . Минимальное значение соответствует точке  $\theta$  близкой к 0.17. Рис. 5б иллюстрирует тот факт, что при работе с фиксированным параметром  $\theta = 0.05$  поведение текущих оценок функционала качества часто приводит к существенному росту функции потерь, что говорит о низкой эффективности обработки очереди заданий. Напротив, использование рассмотренного в работе рандомизированного алгоритма адаптации шага диспетчеризации при тех же стартовых условиях позволяет с течением времени выйти на уровень близкий к среднему минимальному возможному значению, т. е. сервер, начав работу с неоптимальным параметром  $\theta$ , с течением времени адаптируется к конкретным (неизвестным для сервера) параметрам входного потока. Типичный вид поведения значений эмпирического функционала качества при такой подстройке приведен на рис. 6.

На рис. 7 показан пример подстройки настраиваемого параметра сервера к скачкообразному изменению параметров входного потока.

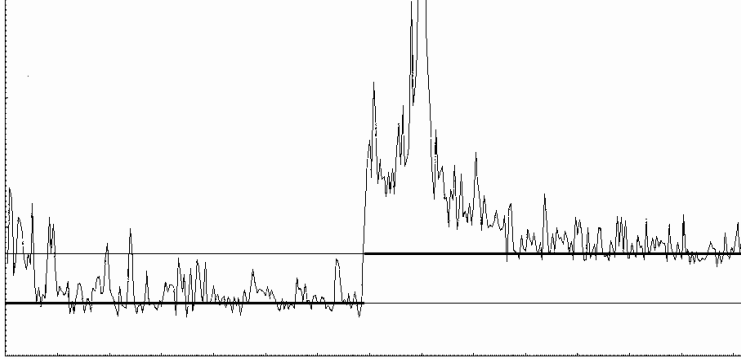


Рис. 7: Поведение  $y_k$  при подстройке к изменению входного потока.

### 3.3. Система управления загрузкой узлов распределенной вычислительной сети

Пусть в системе имеется  $n$  узлов. Шаг работы системы заключается в обработке пакета заданий известного размера  $r_i$ . Предположим, что весь пакет может быть произвольным образом разделен на  $n$  заданий для узлов, при этом узел  $j$  получает задание размером  $u_i^{(j)}$ ,  $j = 1, 2, \dots, n$ ,  $\sum_{j=1}^n u_i^{(j)} = r_i$ . Время вычисления для узла  $j$  определяется по формуле  $t_i^{(j)}(u_i^{(j)}) = \frac{u_i^{(j)}}{\theta_i^{(j)}}$ , где  $\theta_i^{(j)} \in \mathbb{R}$  — производительность узла  $j$ . Необходимо минимизировать по  $u_i = (u_i^{(1)}, u_i^{(2)}, \dots, u_i^{(n)})^T$  среднее время вычисления задания пакета  $r_i$ :

$$Z_i = \frac{T(u_i)}{r_i} = \max_{j=1, \dots, n} \frac{t_i^{(j)}(u_i^{(j)})}{r_i} \rightarrow \min_{u_i}.$$

При введенных обозначениях, если производительности узлов известны и не меняются  $\theta_i \equiv \theta$ , тогда оптимальным является правило пропорционального распределения заданий.

На практике производительности узлов могут быть искажены из-за сторонних заданий, т. е.  $\theta_i = \theta + \bar{x}_i - x_i$ , либо изменяться со временем:  $\theta_i = \theta_{i-1} + \xi_i$ , где  $\xi_i, x_i \in \mathbb{R}^n$  — векторы независимых случайных величин, распределенных по Парето.

Для идентификации или отслеживания изменений  $\theta_i$ , целесооб-



разно использовать контур обратной связи, в который включен РАСА. Для оценивания отклонения среднего времени на выполнение одного пакета заданий от оптимального при использовании РАСА в стационарной и нестационарной задаче в качестве наблюдений можно использовать величины

$$y_i = \sum_{j=1}^n \left( \frac{u_i^{(j)}}{t_i^{(j)}} - \tilde{\theta}_i^{(j)} \right)^\rho,$$

которые в силу выбранной стратегии управления являются зашумленными измерениями функции  $f_i(\theta) = \|\theta_i - \theta\|_\rho^\rho$ . Подробности можно найти в [20]

#### 4. Кодирование/декодирование информации

В XX в. слово “информация” стало популярным термином во многих научных областях, получив особые для каждой из них определения и толкования [21]. Одно из них можно найти в Википедии: *информацией* (от латинского informatio — “научение”, “сведение”, “оповещение”) называется опосредованный формами связи результат отражения изменяемого объекта изменяющимся с целью сохранения их системной целостности. Обычно путают или отождествляют термин “информация” с близкими понятиями “данные” (получаемые сообщения) и “знания”. Понимая многообразность возможных толкований, не углубляясь в детали, в контексте этой статьи будем полагать, что *знания* обеспечивают восстановление необходимой субъекту *информации*  $x$  по получаемым в ходе экспериментов или вычислений *данным*  $y$ . Достаточно часто можно упрощенно считать, что существенная информация об исследуемом явлении  $x \in X$  связана с имеющимися данными  $y \in Y$  через понимание закономерностей явления — знание (оператор)  $\Phi : X \rightarrow Y$ :

$$y = \Phi x \quad (= \Phi(x)).$$

Если оператор  $\Phi$  обратим, то он обеспечивает исчерпывающие знания для полного восстановления  $x$  по  $y$ . При линейной зависимости  $y, x \in \mathbb{R}^N$  и невырожденной  $N \times N$  матрице  $\Phi$  из матричной алгебры хорошо известно, что  $x = \Phi^{-1}y$ .

Для открытых систем типичным является случай, когда данные подвержены влиянию неконтролируемых возмущений

$$y = \Phi x + \xi.$$

Выделение слагаемого  $\xi$  с философской точки зрения подчеркивает влияние на данные  $y$  других явлений вне круга явлений, определяемых  $x$ . Даже при отсутствии прямого внешнего воздействия последнее соотношение — более естественное с практической точки зрения. Обычно получение данных — процесс взаимодействия исследуемого явления (отражаемого информацией  $x$ ) с некоторой измерительной системой со своими характеристиками, совокупность которых объединяет  $\xi$ .

При незначительном уровне внешних возмущений  $\xi$  (или при их затухающем характере) задача о восстановлении  $x$  по  $y$  сводится к проблеме обращения оператора  $\Phi$ , что обычно достигается за счет увеличения количества наблюдений: при  $x \in \mathbb{R}^N$  выбирают  $m > N$  и  $y \in \mathbb{R}^m$ .

При существенных внешних возмущениях  $\xi$  обычно используется статистическая постановка задачи, возможности решения которой при  $m \gg N$  детально изучены в рамках традиционной математической теории планирования экспериментов [22]. Внешние возмущения считают реализацией некоторой последовательности независимых случайных величин с нулевым средним значением. Однако в приложениях это допущение часто нарушается, что может сильно сказываться на работе традиционных оценочных процедур.

На первый взгляд, это кажется удивительным, но задача о восстановлении  $x$  может быть эффективно решена и в случае нецентрированных, коррелированных и даже неслучайных помех [5, 23] за счет случайного выбора матрицы  $\Phi$ . Идея использования случайных регрессоров для устранения эффекта смещения была выдвинута еще Р. Фишером [24] в виде рандомизированного принципа планирования эксперимента. Помимо задачи планирования эксперимента, в которой регрессоры могут быть рандомизированы экспериментатором, случайные входы возникают во многих задачах управления, идентификации, фильтрации, распознавания и т. д.

В [5] делаются общие выводы о том, что **рандомизация** процесса измерений позволяет

- устранить эффект смещения и
- уменьшить количество итераций, а значит, и наблюдений.

В XXI в. резко увеличились объемы обрабатываемой информации. В значительной степени это связано с массовым переходом к обработке потоков двумерных (2-D) и трехмерных (3-D) данных. Сложность традиционных методов квантования сигналов возрастает по экспоненциальному закону с ростом размерности. Квантование 1-D сигналов при  $N = 10^3$  отсчетов соответствует  $10^6$  в случае 2-D, а в 3-D —  $10^9$ , что уже чрезвычайно велико. В современных приложениях для цифровых фото- и видеокамер традиционное требование о необходимой частоте (скорости, Nyquist Rate) измерения настолько высоко, что слишком большое количество получающихся данных надо существенно сжимать перед хранением или пересылкой. В других приложениях, включая системы отображения (медицинские сканеры, радары) и быстродействующие аналого-цифровые конвертеры, увеличение частоты измерений оказывается очень дорогостоящим.

С практической точки зрения чрезвычайно интересно исследовать возможности восстановления  $x \in \mathbb{R}^N$  по  $y \in \mathbb{R}^m$  при  $m \ll N$ , что, конечно же, нереализуемо в общем случае. Но в последнее время на помощь традиционной теории обработки сигналов приходит новая парадигма **Compressive Sensing** (“опознание со сжатием”), позволяющая достаточно точно восстанавливать “разреженную” (sparse) информацию  $x$  [25, 26]. Англоязычный термин *compressive sensing* за последние пять лет стал уже общепотребимым. В [27] предложен перевод на русский язык — *сжатые измерения*. Б.Т. Поляк на одном из научных семинаров в ИПУ РАН предложил использовать термин *опознание*, который, по нашему мнению, точнее отражает специфику нового подхода.

Новая парадигма основана на сохраняющих структуру сигнала неадаптивных линейных проектированиях, по результатам которых информация восстанавливается, например, на основе применения методов  $\ell_1$ -оптимизации:  $x$  находится как решение некоторой задачи вида

$$\|x\|_1 = \sum_j |x[j]| \rightarrow \min : y = \Phi x.$$

Новая методика базируется на определенном (обычно рандомизированном) выборе матрицы  $\Phi$  и на том, что получающийся в результате  $\ell_1$ -оптимизации вектор  $\hat{x}$  имеет не более  $m$  ненулевых компонент, т. е. он сильно разрежен. Последний замечательный факт был установлен в студенческой курсовой работе одного из авторов

при решении проблемы о построении  $\ell_1$  оптимального стабилизирующего регулятора неминимальнофазового объекта, первоначально представленной в [28]. В 1984 г. в журнале “Автоматика и телемеханика” было дано с детальной геометрической интерпретацией развернутое объяснение того, почему у вектора решения задачи  $\ell_1$ -оптимизации не равны нулю только малое количество компонент (равное коразмерности подпространства  $\Phi x = 0$ ) [29]. Через три года похожий результат был опубликован в [30].

Более детальное описание новой парадигмы и много примеров можно найти в [31–33].

## 5. Новые модели вычислительных процессов

Хорошо известно, что компьютеры становятся все миниатюрнее и миниатюрнее. Когда-то предполагали, что более мощные машины будут требовать больше места под периферию, память и т.д. Это предположение оказалось неверным. В 1965 г. Г. Мур сформулировал действующее и сейчас правило (впоследствии названное законом Мура), согласно которому производительность вычислительных систем удваивается каждые восемнадцать месяцев. Г. Мур вывел свой эмпирический закон, построив зависимость числа транзисторов в интегральной микросхеме от времени. Как следствие, из этого закона можно вывести темпы миниатюризации отдельного транзистора. Развитие цифровых электронных технологий приводит к тому, что размер элементарного вычислительного устройства приближается к размеру молекулы или даже атома. На таком уровне законы классической физики перестают работать и начинают действовать квантовые законы, которые для многих важных динамических задач еще не описаны теоретически. Увеличение быстродействия вычислительных устройств и уменьшение их размеров с неизбежностью приводит к необходимости операций с “переходными” процессами. Вместо примитивных операций с классическими битами в будущем было бы естественно перейти операциям, задаваемыми теми или иными динамическими моделями микромира. Под примитивной “моделью” вычислений можно понимать выполнение классических операций с битами. Конечно же введение более широкого класса моделей было бы более обоснованным, если бы удалось, например, для функции, значения которой записаны в кластере квантовых битов, определить операцию, эффективно вы-

полняющую преобразование Фурье. Причем может оказаться, что время на ее выполнение будет вполне соизмеримым со временем выполнения одной классической операции, так как операции типа “свертки” функций вполне могут обнаружиться “в природе”. Последние исследования похожих моделей показывают, что их выполнение за счет присущей природе способности к самоорганизации не обязательно “раскладывается” на более простые кирпичики, т. е. не всегда может быть записано в виде классического алгоритма.

В информатике в рамках теории сложности алгоритмов получен целый ряд результатов, обосновывающих теоретическую невозможность эффективного решения многих значимых с практической точки зрения задач, в частности, связанных с моделированием естественных процессов. Это означает, что при попытке описания интересующего нас процесса с заданной точностью при помощи машины Тьюринга ее алфавиты состояний и памяти и/или время функционирования оказываются недопустимо велики. Одними из первых с этими теоретическими трудностями столкнулись разработчики систем автоматического управления, которые ведут исследования на стыке информатики и той или иной конкретной научной дисциплины (физики, биологии, химии и т. п.). Реализации целого ряда современных методов теории автоматического управления не укладываются в рамки дискретной модели традиционной машины Тьюринга. Один из возможных способов ее обобщения — это использование так называемых “гибридных стохастических систем”, свойства которых активно изучаются в последнее время.

Одним из вариантов расширения понятия вычислимости может стать введение нового понятия вычислительного устройства, в котором вычисления строятся не на традиционных алгоритмах, а на результатах работы асинхронных динамических моделей. Это позволит переосмыслить понятие “такт”, противоречивый смысл выражается в рамках неразрешимости проблемы континуума в рамках аксиоматики Френкеля-Цермело, и отказаться от другого серьезного ограничения классической модели — разбиения памяти на изолированные биты. Последнее особенно важно по двум причинам. Во-первых, сокращение длины такта и расстояний между битами с определенного уровня делает невозможным рассматривать их изолированно в силу законов квантовой механики. Во-вторых, отказ от “скалярных битов” позволит говорить о выполнении многомерных (векторных) операций за один “такт”. Например, П. Шор [34] пред-

ложил алгоритм квантового преобразования Фурье, которое может выполняться за время пропорциональное  $(\log_2 N)^2$ , а не за  $N \log_2 N$  как классическое быстрое преобразование Фурье. В [35] обсуждается опирающийся на квантовую адиабатическую теорему гипотетически возможный “физический” способ решения за конечное время 10-ой проблемы Гильберта, в [15] предложен эффективный квантовый алгоритм вычисления “за такт” оценки вектора-градиента многомерной функции.

В некотором смысле переход к разработке и созданию моделей сложных вычислений — закономерный этап развития микропрограммирования. Понимание возможных преимуществ работы устройств с эффективным микрокодом отмечалось еще в 50-е гг. прошлого века. Во втором ряду серии ЕС ЭВМ в конструкцию была заложена возможность динамического микропрограммирования. Хорошо известен целый ряд ЭВМ от “Мир-1” до “Эльбрус”, использующих языки высокого уровня (HLL) как машинные. На кафедре системного программирования СПбГУ легендой стал компьютер “САМСОН”, при создании которого удалось разработать эффективную технологию микропрограммирования [36]. В настоящее время *технологии микропрограммирования естественным образом сдвигаются в сторону физических процессов.*

### 5.1. Асинхронные модели динамических систем

В [37, 38] была предложена новая концепция вычислительного устройства, обобщающая схему классической машины Тьюринга. В рамках новой модели переосмысливаются ставшие традиционными понятия “такта”, “памяти”, “программы” и “состояния”. В частности, в новой модели “память” представляет собой модель динамической системы, “тактом” считается достижение терминальных множеств, а “программа” — “скачкообразную” эволюцию памяти по достижении терминальных множеств. Новая концепция принципиально отличается от классической включением “непрерывности” эволюции.

Другим дополнительным обобщением машины Тьюринга может быть вероятностное задание отображений эволюции и программ, что позволит реализовывать с помощью новой модели динамические системы, стохастические гибридные системы, вероятностные автоматы, системы со стохастическим управлением и т. п., не описываемые детерминированными законами. Кроме того, в [39]

показан пример эффективности использования рандомизированных программных воздействий на систему в условиях динамически изменяющейся структуры пространства состояний. Рандомизация позволяет частично устранить влияние на работу системы систематических погрешностей [5, 23], которые практически неизбежны при изменяющейся со временем модели динамической системы.

В любых современных вычислительных устройствах хранение информации основано на тех или иных физических принципах, только эволюция состояния ячейки во время хранения более проста — сохраняется постоянной какая-то физическая характеристика. Другими словами, традиционную информатику можно сравнить с арифметикой, она оперирует цифрами — значениями ячеек памяти (или, более точно — арифметикой конечных двоичных дробей). Предлагаемая новая модель вычислений позволит перейти в информатике от “арифметики” к “функциональному анализу”, исследующему процессы эволюции информации внутри новых “ячеек”.

Предложенная новая модель вычислений позволяет описывать если не все, то подавляющее большинство процессов реального мира, а также работу всевозможных существующих и будущих вычислительных устройств, включая аналоговые и биокомпьютеры, нейромкомпьютеры, квантовые компьютеры и т. д. Особенностью предлагаемого подхода является отказ от редукции сложности в процессе вычисления. Сложность вычислимого объекта должна быть эквивалентна сложности вычисляемого. Таким образом, понятие вычислительной сложности правильнее рассматривать относительно выбранной системы базисных эволюционных примитивов, а не относительно традиционно рассматриваемых битовых преобразований  $\{0, 1\}$ . Квантовые и нейромкомпьютеры обещают сильно изменить представления о вычислительной мощности современных вычислительных устройств. Увеличение вычислительной мощности, возможное за счет использования новых моделей вычислений, основывающихся на физических явлениях, позволяет предположить, что в будущем новые компьютеры смогут решать задачи, невыполнимые для обычных компьютеров.

## 5.2. Искусственный интеллект

За более чем пятидесятилетнюю историю практического использования компьютеров одним из основных разочарований для человечества стала невозможность создания искусственного интеллекта. Несмотря на внушительный прогресс в сфере создания мощных вычислительных устройств, пока еще нельзя говорить о детальном понимании этой задачи или о направлениях исследований, которые могли бы дать результат.

Большая часть работ по искусственному интеллекту, известных автору, посвящена решению конкретных задач, или реализациям каких-либо эвристических поведенческих алгоритмов, эффективно справляющихся с тестовыми заданиями и терпящих неудачу на задачах реального мира.

Перед тем как перейти к обсуждению подходов создания искусственного интеллекта, сделаем небольшой обзор достижений в математике и информатике за прошедший век, что позволит нам с большим оптимизмом смотреть на возможность решения обозначенной проблемы в будущем. Первые компьютеры были специализированы для решения конкретных задач. Это обосновывалось экономическими причинами и слабой электротехнической базой. За короткое время технический прогресс позволил создать универсальный компьютер, а теоретические исследования позволили эффективно решать большое количество поставленных задач. Окончательно концепция универсального языка программирования закрепились в середине шестидесятых, и последние четыре десятилетия технология программирования бурно развивалась. Несмотря на это, остались задачи, эффективное решение которых требует невероятных вычислительных мощностей (например, обращение матриц, факторизация больших чисел, преобразования Фурье и Лапласа и т. д.). Многие из них формируют минимальную базу для построения различных интеллектуальных систем.

Сегодня наилучший способ построения быстрых вычислительных устройств заключается в организации параллельных вычислений для переработки больших объемов информации. Наиболее перспективны здесь два направления. Первое связано с созданием классических электронных устройств, таких как VLSIC (Very Large Scale Integrated Circuits). На данный момент уже доступны технологии от “35 нм”. Процессоры, построенные по этим технологиям



способны одновременно выполнять сотни тысяч операций. Компания IBM объявила, что в течении нескольких лет будет создано устройство, выполняющее одновременно миллион вычислительных операций. Такие вычислительные мощности позволят быстро обрабатывать матрицы больших размеров, что необходимо, например, в задачах, связанных с распознаванием.

Второе, более перспективное направление в создании вычислительных устройств связано с использованием квантовых компьютеров. Для решения определенных видов задач, наверное, будут создаваться соответствующие молекулы и квантовые алгоритмы.

Несмотря на широкую распространенность специализированных параллельных процессоров многие исследователи согласны с тем, что пока еще не разработана методология создания алгоритмов и устройств для решения задач искусственного интеллекта. Разработка подобной методологии по всей видимости связана с анализом различных задач искусственного интеллекта, выявлением аналогичных частей и созданием новой формальной логики, адаптированной под эти части.

В [40] был предложен один из возможных подходов к созданию интеллектуальной системы. Термин “интеллектуальная” подразумевает способность системы адаптироваться к меняющемуся окружению посредством выбора задачи, которую необходимо решить в текущий момент. Такая система может состоять из двух частей: внутренней, и внешней, регулирующей взаимодействие с внешним миром (датчики, сенсоры, манипуляторы, гусеницы, и т. д.). Внутренняя часть должна удовлетворять следующим условиям:

- для каждой из решаемых системой задач имеется соответствующее устройство, способное оптимально решать эту задачу,
- информация из внешнего мира поступает на входы всех устройств одновременно.

При выполнении этих условий все устройства будут интерпретировать и обрабатывать информацию из реального мира параллельно.

Можно вообразить множество путей реализации такой системы. Далее разберем один из них. Предположим, что нужна система, способная решать некоторое конечное множество задач:

$$\{A_1, A_2, \dots, A_M\}.$$

Здесь  $M$  — натуральное число (1, 2, 3 или 100000, 1000000). Представим, что наши устройства реализованы в виде молекул (квантовых компьютеров), взаимодействующих друг с другом и “свернутых”, например, в спираль или шар

$$\{D_1, D_2, \dots, D_M\}.$$

Подобная пространственная форма может облегчить одновременность для всех устройств некоторого биологического или физического воздействия. Положим, что поведение каждого из устройств определяется его внутренней структурой, потоком данных из внешнего мира и некоторым конечным набором параметров  $\theta^{(i)}$  из множества  $\Theta_i$

$$D_i = D_i(\theta^{(i)}), \theta^{(i)} \in \Theta_i, i = 1, 2, \dots, M.$$

Все параметры системы в этом случае могут быть перечислены следующим образом:

$$\theta = (\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(M)})^T, \theta \in \Theta = \Theta_1 \times \dots \times \Theta_M.$$

Пусть эта схема удовлетворяет обоим перечисленным выше условиям.

Более прост в реализации вариант, в котором каждое из устройств представлено отдельной нейронной сетью. Теоретический аппарат нейронных сетей уже достаточно сильно проработан (по сравнению с квантовыми компьютерами), что дает возможность строить систему из готовых кирпичиков. Недостатком такого подхода является необходимость размножения входных данных для всех устройств системы.

Приведем для примера задачу, для решения которой подобная схема представляется подходящей. Предположим, что у нас есть набор устройств, способных распознавать определенные виды изображений (каждое из устройств настроено на какой-либо конкретный объект). Все эти устройства получают в реальном времени изображение с некоторой цифровой камеры. В каждый момент времени только одно из устройств, объявившее себя главным, может управлять камерой. Существует также одно устройство, управляющее камерой в случаях, когда никто не объявил себя главным. Оно может действовать по какому-нибудь простому алгоритму — например, осматриваться или сканировать некоторый важный участок, в

котором ожидается появление объектов. Картинка с камеры постоянно подается на входы всех распознавателей, и те в свою очередь пытаются распознать в ней свой объект. Результатом их работы является степень совпадения в процентах (например, объект является кошкой с вероятностью 75%). Если у какого-то из устройств степень достоверности превысила некий заранее установленный порог (например, 90%), то это устройство объявляет себя главным и берет на себя управление камерой для более детального изучения объекта или слежения за ним.

Здесь мы подошли к одной из самых важных концепций предлагаемого решения — понятию *информационного резонанса* [40]. Определим его следующим образом — устройство входит в резонанс с входными данными, если оно распознало в них корректные входные параметры и может успешно решать задачу. В системе неизбежно будут возникать три ситуации:

- в резонанс вошло только одно устройство,
- в резонанс вошли несколько устройств,
- ни одно из устройств не вошло в резонанс.

Первый случай уже был описан ранее. Во втором можно выбрать главное устройство по каким-либо показателям силы (качества) резонанса (в последнем примере — по величине показателя достоверности распознавания). Третий случай наиболее интересен. Он может, например, означать, что параметры устройств надо настроить для получения резонанса. Возникает задача многомерной оптимизации:

$$f(\theta) = 1 - \max(\text{prob}_i(\theta^{(i)})) \rightarrow \min_{\theta}.$$

В этом случае минимум функции потерь достигается, когда максимально возможное качество распознавания получается на одном из устройств. Для решения поставленной задачи предлагается рассмотренный выше в этой статье рандомизированный алгоритм стохастической оптимизации, который хорошо ложится на логику квантовых вычислительных устройств и по многим параметрам предпочтительнее других подобных алгоритмов для оптимизации систем в реальном времени.

Рассмотрим пример интеллектуальной системы, работающей на перечисленных выше принципах. Представим себе робота, взаимодействующего с реальным миром через стандартные устройства

— датчики, камеры, микрофоны, радиолокаторы, гусеницы, шупальца, и т. д. Центральный компьютер робота состоит из тысяч устройств, обрабатывающих одни и те же данные, поступающие из внешнего мира, и создающих внутреннее представление мира в мозгу робота. Это внутреннее представление по очевидным причинам будет создаваться с помехами и постоянно корректироваться.

Центральный компьютер использует созданное представление мира для расчета оптимальной траектории, позволяющей роботу избежать опасностей и выполнить поставленное задание. Функционирование робота в подобных условиях можно представить как одновременное выполнение нескольких параллельных задач, контролируемых разными устройствами в центральном компьютере. Следует отметить, что если резонансные устройства не конкурируют за общий ресурс (камеру, руку и т. д.), то нет необходимости выбирать из них главное.

Работа всех устройств робота (в том числе и вычислительных) регулируется вектором параметров, размерность которого может быть очень большой. Поиск оптимальных значений этих параметров возможен только после постановки задачи, включающей в себя определение функционала качества. В данном случае функционал качества будет оцениваться центральным компьютером. Основные возможные критерии — соответствие внутренней картины мира реальным условиям (число коррекций), число и качество информационных резонансов и т. д. Настройку параметра с высокой размерностью удобнее всего производить описанным выше методом.

## 6. Обработка изображений

В дипломных работах Д.В. Павленко “Рандомизированные алгоритмы в задаче поиска образца в изображении” (2009), И.С. Минина “Автоматическое определение и отслеживание целей по сигналу морского радара” (2010) и В.А. Никитина “Применение подхода Compressive Sensing в задаче кодирования лиц” (2010) рандомизированные алгоритмы стохастической аппроксимации использовались для поиска изменений в последовательных кадрах данных видеонаблюдений, морских радаров и для кодирования лиц людей.

## 7. Применения в финансовой математике

В 2009-10 гг. под руководством О.Н. Граничина было выполнено несколько работ, в которых рандомизированные алгоритмы стохастической оптимизации использовались для оценивания параметров финансовых моделей. Е. Варфоломеева “Аналитический модуль для прогнозирования доходности финансовых инструментов на основании GARCH-модели волатильности” (2009) и В. Федяшов “Разработка и реализация модуля прогнозирования волатильности с использованием рандомизированных алгоритмов” (2010) изучали возможности оценивания параметров изменения волатильности, А. Гачков “Платформа для разработки, тестирования и сравнения финансовых индикаторов” (2009) применил рандомизированный подход к оцениванию фрактальной размерности временных рядов, А.Д. Котельников “Применение имитационного моделирования для получения робастного дерева решений инвестиционного проекта” (2009) использовал идеи рандомизированного обучения при анализе инвестиционных проектов, А. Зайцев “Проектирование информационной системы поддержки принятия решения, обеспечивающей максимизацию валовой прибыли” (2010) разработал оригинальную модификацию алгоритма для оптимального формирования цен в супермаркете.

## 8. Адаптивное мультиагентное управление БПЛА

Основные усилия исследовательской группы профессора Граничина О.Н. в настоящее время фокусируются на возможных приложениях изложенных выше идей и методов в задачах адаптивного мультиагентного управления группой беспилотных летательных аппаратов (БПЛА) [41]. Первые результаты уже получены в 2010 году в дипломных работах Е.И. Антал “Оптимизация энергопотребления группы БПЛА с использованием восходящих воздушных потоков” и Леви Сергей “Разработка мультиагентной системы для управления группой БПЛА” [42, 43].

## Заключение

В этой статье приведены основные направления научных исследований в группе профессора О.Н. Граничина, но полный список

тем описанными не исчерпывается. Более подробную информацию можно найти на веб-сайте [www.math.spbu.ru/user/gran](http://www.math.spbu.ru/user/gran). Интересные новые темы будут возникать из обсуждений, дискуссий и конкретных практических проектов.

## Список литературы

- [1] *Vidyasagar M.* Statistical Learning Theory and Randomized Algorithms for Control // IEEE Control Systems. 1998. No 12. P. 69–85.
- [2] *Metropolis N., Ulam S.* The Monte Carlo Method // J. Amer. statistical assoc. 1949. Vol. 44. No. 247. P. 335–341.
- [3] *Calafiore G.C., Campi M.C.* The scenario approach to robust control design // IEEE Trans. Automat. Control. Vol. 51. No. 5. May 2006. P. 742–753.
- [4] *Граничин О.Н., Халидов В.И.* Рандомизированный подход к обнаружению разрывов функции // Стохастическая оптимизация в информатике. Вып. 1. 2005. С. 73–80.
- [5] *Граничин О.Н., Поляк Б.Т.* Рандомизированные алгоритмы оценивания и оптимизации при почти произвольных помехах. — М.: Наука. 2003. 293 с.
- [6] *Граничин О.Н., Измакова О.А.* Рандомизированный алгоритм стохастической аппроксимации в задаче самообучения // Автоматика и телемеханика. 2005. №8. С. 52–63.
- [7] *Avros R., Granichin O., Shalymov D., Volkovich V., Weber G.-W.* Randomized Algorithm of Finding the True Number of Clusters Based on Chebychev Polynomial Approximation. / Chapter of the book on Data Mining D. Holmes (ed.). — Springer. 2010.
- [8] *Avros R., Granichin O., Shalymov D., Volkovich V.* A randomized algorithm for estimation number of clusters // In Proc. of the 24th European Conference on Operation Research. July 11-14. 2010. Lisbon, Portugal.

- [9] *Sugar C., James G.* Finding the number of clusters in a data set : An information theoretic approach // J. of the America Statistical Association. **98**. 2003. P. 750–763.
- [10] *Граничин О.Н., Халидов В.И.* Рандомизированный подход к обнаружению разрывов функции // Стохастическая оптимизация в информатике. 2005. Вып. **1**. С. 73–80.
- [11] *Граничин О. Н.* Об одной стохастической рекуррентной процедуре при зависимых помехах в наблюдении, использующей на входе пробные возмущения // Вестник Ленингр. ун-та. Сер. 1. 1989. №1(4). С. 19–21.
- [12] *Spall J.C.* Multivariate Stochastic Aproximation Using a Simultaneous Perturbation Gradient Aproximation // IEEE Trans. Automat. Contr. 1992. Vol. **37**. No. 3. P. 332–341.
- [13] *Поляк Б. Т., Цыбаков А. Б.* Оптимальные порядки точности поисковых алгоритмов стохастической аппроксимации // Проблемы передачи информации. 1990. **26**. С. 126–133.
- [14] *Chen H.-F., Duncan T. E., Pasik-Duncan B.* A Kiefer-Wolfowitz algorithm with randomized differences // IEEE Trans. Automat. Contr. 1999. Vol. **44**. No. 3. P. 442–453.
- [15] *Вахитов А.Т., Граничин О.Н., Сысов С.С.* Точность оценивания рандомизированного алгоритма стохастической оптимизации // Автоматика и телемеханика. 2006. № 4. С. 86–96.
- [16] *Вахитов А.Т., Граничин О.Н., Гуревич Л.С.* Алгоритм стохастической аппроксимации с пробным возмущением на входе в нестационарной задаче оптимизации // Автоматика и телемеханика. 2009. № 11. С. 70–79.
- [17] *Kushner H.J., Yin G.G.* Stochastic Approximation Algorithms and Applications. — New York: Springer-Verlag. 2002. 415 p.
- [18] *Tang Q-Y., Chen H.F., Han Z-J.* Convergence rates of Perturbation-Analysis-Robbins-Monro-Single-Run algorithms for single server queues // IEEE Trans. on Automatic Control. 1997. Vol. **42**. № 10. P. 1442–1447.

- [19] *Волкович Я.В., Граничин О.Н.* Адаптивная оптимизация сервера, обрабатывающего очередь заданий // Стохастическая оптимизация в информатике. Вып. 1. 2005. С. 17–28.
- [20] *Вахитов А.Т., Граничин О.Н., Паньшенсков М.А.* Методы оценивания пропускной способности в распределенных системах // Нейрокомпьютеры: разработка, применение. 2009. № 11. С. 45–53.
- [21] *Граничин О.Н., Кияев В.И.* Информационные технологии в управлении. — М.: Изд-во Бином. 2008. 336 с.
- [22] *Ермаков С.М., Жиглявский А.А.* Математическая теория оптимального эксперимента. — М.: Наука, 1987.
- [23] *Granicin O.N.* Linear regression and filtering under nonstandard assumptions (Arbitrary noise) // IEEE Trans. Automat. Contr. Oct. 2004. Vol. 49. P. 1830–1835.
- [24] *Fisher R.A.* The Design of Experiments. — Edinburgh: Oliver and Boyd, 1935.
- [25] *Candes E., Romberg J., Tao T.* Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information // IEEE Trans. Inform. Theory. Feb. 2006. Vol. 52. No. 2. P. 489–509.
- [26] *Donoho D.* Compressed sensing // IEEE Trans. Inform. Theory. Apr. 2006. Vol. 52. No. 4. P. 1289–1306.
- [27] *Кашин Б.С., Темляков В.Н.* Замечание о задаче сжатого измерения // Мат. заметки. 2007. Т. 82. № 6. С. 829–837.
- [28] *Граничин О.Н.* Оптимальное управление линейным объектом с нерегулярными ограниченными помехами // В сб. Тез. докл. и сообщений Всесоюз. конф. “Теория адаптивных систем и её применения”. М.–Л.:ВИНИТИ. 1983. С. 26.
- [29] *Барабанов А.Е., Граничин О.Н.* Оптимальный регулятор линейного объекта с ограниченной помехой // АиТ. 1984. № 5. С. 39–46.



- [30] *Dahlem M., Pearson J.B.*  $\ell_1$  optimal feedback controllers for MIMO discrete systems // IEEE Trans. Automatic Control. 1987. Vol. **32**. No. 4. P. 314–322.
- [31] *Граничин О.Н.* Рандомизация измерений и  $\ell_1$ -оптимизация // Стохастическая оптимизация в информатике. Вып. **5**. 2009. С. 3–23.
- [32] *Граничин О.Н., Павленко Д.В.* Рандомизация данных и  $\ell_1$ -оптимизация // Компьютерные инструменты в образовании. 2010. №1. С. 5–13.
- [33] *Граничин О.Н., Павленко Д.В.* Рандомизация получения данных и  $\ell_1$ -оптимизация (опознание со сжатием) (Обзор) // Автоматика и телемеханика. 2010. № 11.
- [34] *Shor P.* Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer // SIAM J. Comput. 1997. Vol. **26**. P. 1484–1509.
- [35] *Tien D.K.* Computing the non-computable // Contemporary Physics. Vol. **44**. 2003. P. 51–71.
- [36] *Terekhov A. N.* The main concepts of a new HLL computer “САМСОН” // Computer Science Journal of Moldova. 1993. Vol. **1**. № 1.
- [37] *Граничин О.Н., Жувикина И.А.* Новая концепция процесса вычислений, основанная на эволюционных примитивах // Системное программирование. Вып. **1**. 2005. С. 68–83.
- [38] *Granichin O.N., Vasilev V.I.* Computational model based on evolutionary primitives. Turing machine generalization // International Journal of Nanotechnology and Molecular Computation. Vol. **2**. 2010. No. 2. P. 30–43.
- [39] *Granichin O.N., Khantuleva T.A.* Hybrid systems and randomized measuring in nonequilibrium processes // Differential Equation and Control Processes. 2004. № 3. Electronic J. N P23275 at 07.03.97.
- [40] *Granichin O.N., Sysoev S.S.* About some characteristics of computers of new generation // In Proc. of the Physics and Control Conference. Saint-Petersburg. 2003. Vol. 3. P. 804–807.

- [41] *Амелин К.С., Антал Е.И., Васильев В.И., Граничина Н.О., Граничин О.Н.* Свидетельство о государственной регистрации программы для ЭВМ №2010612684 “SmartFly Together” от 26 февраля 2010 г.
- [42] *Амелин К.С., Антал Е.И., Васильев В.И., Граничина Н.О.* Адаптивное управление автономной группой беспилотных летательных аппаратов // Стохастическая оптимизация в информатике. Вып. 5. 2009. С. 157–166.
- [43] *Antal C., Granichin O., Levi S.* Adaptive autonomous soaring of multiple UAVs using SPSA // In Proc. of the 49th IEEE Conference on Decision and Control, December 15-17, 2010, Atlanta, GA. USA.