

ISSN 2222-8772

**М**АТЕМАТИЧЕСКИЕ  
**С**ТРУКТУРЫ  
И  
**М**ОДЕЛИРОВАНИЕ

Том 27, №1  
2013



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМ. Ф.М. ДОСТОЕВСКОГО»**

**МАТЕМАТИЧЕСКИЕ  
СТРУКТУРЫ  
И  
МОДЕЛИРОВАНИЕ**

**Том 27, № 1**

Омск  
2013

**Математические структуры и моделирование.** — Омск : Омский государственный университет, 2013. — Т. 27, № 1. — 122 с.

ISSN 2222-8772 (print)

ISSN 2222-8799 (online)

---

### Редакционная коллегия

---

<b>А. А. Берс</b>	доктор техн. наук, профессор, Институт систем информатики СО РАН им. А. П. Ершова. Новосибирск
<b>Н. Ф. Богаченко</b>	канд. физ.-мат. наук, доцент, Омский государственный университет им. Ф. М. Достоевского
<b>В. Я. Волков</b>	доктор техн. наук, профессор, зав. кафедрой начертательной геометрии, инженерной и машинной графики, Сибирская государственная автомобильно-дорожная академия (СибАДИ)
<b>А. Г. Гринь</b>	доктор физ.-мат. наук, профессор, кафедра кибернетики, Омский государственный университет им. Ф. М. Достоевского
<b>С. И. Горлов</b>	доктор физ.-мат. наук, профессор, Нижневартовский государственный гуманитарный университет
<b>А. К. Гуц</b>	доктор физ.-мат. наук, профессор, зав. кафедрой кибернетики, Омский государственный университет им. Ф. М. Достоевского
<b>А. Н. Кабанов</b>	канд. физ.-мат. наук, кафедра кибернетики, Омский государственный университет им. Ф. М. Достоевского
<b>П. А. Корчагин</b>	доктор техн. наук, профессор, Сибирская государственная автомобильно-дорожная академия (СибАДИ)
<b>Д. Н. Лавров</b>	главный редактор, канд. техн. наук, доцент, зав. каф. компьютерных технологий и сетей, Омский государственный университет им. Ф. М. Достоевского
<b>А. А. Fedorenko</b>	Ph.D., Researcher (CR1) at the French National Centre of Scientific Research (CNRS) Laboratoire de Physique de l'ENS-Lyon, France
<b>V. Kreinovich</b>	Ph.D., Professor, Computer Science Department, University of Texas at El Paso, Texas, USA

---

### Адрес научной редакции

---

Россия, 644053, Омск-53, ул. Грозненская, 11  
Омский государственный университет им. Ф. М. Достоевского  
факультет компьютерных наук  
E-mail: lavrov@omsu.ru

---

## МАТЕМАТИЧЕСКИЕ СТРУКТУРЫ И МОДЕЛИРОВАНИЕ

В журнале публикуются статьи, в которых излагаются результаты исследований по фундаментальной и прикладной математике, теоретической физике и размышления, касающиеся окружающей нас природы и общества.

Публикуются также статьи по информационным технологиям, компьютерным наукам, защите информации, философии и истории математики.

Объекты исследования должны быть представлены в форме некоторых математических структур и моделей.

Журнал является реферируемым. Рефераты статей публикуются в «Реферативном журнале» и в журналах «Zentralblatt für Mathematik» (Германия) и «Mathematical Reviews» (США).

Электронная версия журнала представлена в сети Интернет по адресам:

<http://msm.univer.omsk.su>

<http://msm.omsu.ru>

Журнал издаётся на коммерческие средства факультета компьютерных наук Омского государственного университета.

Электронная почта редактора:

[lavrov@omsu.ru](mailto:lavrov@omsu.ru)

Подробную информацию можно найти на Web-серверах:

<http://msm.univer.omsk.su>

<http://msm.omsu.ru>

# СОДЕРЖАНИЕ

## Фундаментальная математика и физика

- А.Г. Гринь. *О неравенстве Магды Пелиград* . . . . . 5
- А.Н. Романов. *Различающие и глобально гиперболические пространства* . . . . . 11
- A.A. Fedorenko. *Surface criticality in random field systems with continuous symmetry* . . . . . 14
- О. Kosheleva, V. Kreinovich. *Brans-Dicke Scalar-Tensor Theory of Gravitation May Explain Time Asymmetry of Physical Processes* . . 28
- О. Kosheleva, V. Kreinovich. *For Describing Uncertainty, Ellipsoids Are Better than Generic Polyhedra and Probably Better than Boxes: A Remark* . . . . . 38

## Прикладная математика и моделирование

- А.К. Гуц, Д.Н. Лавров. *Описание DDoS-атаки с помощью катастрофы «сборка»* . . . . . 42
- Ю.В. Коваленко. *Модель с непрерывным представлением времени для задачи составления расписаний с группировкой машин по технологиям* . . . . . 46

## Компьютерные науки

- С.В. Гусс. *Концепция проекта расширяемого обучающего программного средства* . . . . . 56
- С.В. Стопкин, К.В. Пугин, Д.М. Бречка. *Установка программного обеспечения на вычислительный модуль с архитектурой процессора Alpha* . . . . . 75
- Е.А. Тюменцев. *Автоматизированное тестирование сложности алгоритмов с помощью Моск-объектов* . . . . . 82

Продолжение на следующей странице



### Информационная безопасность

- А.А. Коровянский. *Обзор решений для реализации прототипа замка мобильной биометрической системы доступа M-Block* ..... 89
- Р.С. Прохоров. *Бихевиористическая идентификация процессов* ..... 103

### Страничка памяти

- Андрей Александрович Берс. *1934-2013* ..... 113

## О НЕРАВЕНСТВЕ МАГДЫ ПЕЛИГРАД

А.Г. Гринь

Предлагается новое доказательство вариантов известного неравенства М. Пелиград, в том числе, использующих условия слабой зависимости, отличные от  $\varphi$ -перемешивания.

Пусть  $\{\xi_n\} = \{\xi_n, n = 1, 2, \dots\}$  — стационарная в узком смысле последовательность и пусть  $\mathcal{F}_{\leq n}$  и  $\mathcal{F}_{\geq n}$  —  $\sigma$ -алгебры, порождённые семействами  $\{\xi_i : i \leq n\}$  и  $\{\xi_i : i \geq n\}$ . Говорят, что последовательность  $\{\xi_n\}$  удовлетворяет условию равномерно сильного перемешивания ( $\varphi$ -перемешивания) с коэффициентом перемешивания  $\varphi(n)$ , если

$$\varphi(n) = \sup \left\{ \frac{|\mathbb{P}(AB) - \mathbb{P}(A)\mathbb{P}(B)|}{\mathbb{P}(A)} : A \in \mathcal{F}_{\leq 0}, B \in \mathcal{F}_{\geq n} \right\} \rightarrow 0, \quad n \rightarrow \infty.$$

Обозначим

$$S_0 = 0, \quad S_n = \sum_{k=1}^n \xi_k, \quad T_n = \max_{1 \leq k \leq n} |S_k|, \quad n = 1, 2, \dots \quad (1)$$

**Теорема 1.** (Неравенство М. Пелиград)

Пусть при некотором  $a > 0$  и натуральном  $m$

$$\varphi(m) + \max_{1 \leq j \leq m} \mathbb{P}\{|S_j| \geq a\} \leq \gamma < 1.$$

Тогда при любых  $x > 0$ ,  $y \geq a$  и  $n \geq m$

$$\mathbb{P}\{|T_n| > x + 4y\} \leq \frac{\gamma}{1 - \gamma} \mathbb{P}\{|T_n| > x\} + \frac{1}{1 - \gamma} \mathbb{P}\left\{ \max_{1 \leq i \leq n} |\xi_i| > \frac{y}{m} \right\}.$$

Теорема 1 получена Магдой Пелиград в [1]. «Изюминка» этого неравенства в том, что параметр  $\gamma$  можно сделать сколь угодно малым и хвосты распределений сумм и максимумов становятся в некотором смысле похожими. Из этого факта выводятся разнообразные следствия (равносильность равномерной интегрируемости степеней сумм и максимумов, неулучшаемые по порядку оценки

моментов сумм слабо зависимых величин и т. д.), которые стимулировали существенный прогресс в предельных теоремах для слабо зависимых величин. К примеру, достигнуты серьёзные продвижения в доказательстве так называемой гипотезы Ибрагимова-Иосифеску [1, 2], получены необходимые и достаточные условия для притяжения стационарных последовательностей с  $\varphi$ -перемешиванием к устойчивым законам [3–5] и т. д.

В настоящей заметке предлагается новое доказательство неравенств типа неравенства М. Пелиград, в которых помимо  $\varphi$ -перемешивания используются некоторые другие условия слабой зависимости.

Пусть  $\{\xi_n\}$  — последовательность случайных величин. Введём  $S_n$  и  $T_n$  по формулам (1) и обозначим

$$Q_n = \max_{1 \leq k \leq n} \min \{|S_k|, |S_n - S_k|\}, \quad Q_n(m) = \max_{m \leq k \leq n} \min \{|S_{k-m}|, |S_n - S_k|\},$$

$$(Q_n(0) = Q_n).$$

**Лемма 1.**

$$T_n \leq 3Q_n(m) + (3m + 1) \max_{1 \leq i \leq n} |\xi_i|. \quad (2)$$

Доказательство основано на идеях из [6, гл.2, §12]

Пусть  $M = \{k : |S_k| \leq |S_n - S_k|\}$ ; ясно, что  $0 \in M$ . Пусть  $S_n \neq 0$ . Тогда  $n \notin M$  и, следовательно, существует  $0 < k \leq n$  такое, что  $k - 1 \in M$  и  $k \notin M$ , то есть

$$|S_{k-1}| \leq |S_n - S_{k-1}|, \quad |S_k| > |S_n - S_k|.$$

Тогда

$$\begin{aligned} |S_n| &\leq |S_{k-1}| + |\xi_k| + |S_n - S_k| \leq \min \{|S_{k-1}|, |S_n - S_{k-1}|\} + \\ &+ \min \{|S_k|, |S_n - S_k|\} + \max_{1 \leq i \leq n} |\xi_i| \leq 2Q_n + \max_{1 \leq i \leq n} |\xi_i|. \end{aligned} \quad (3)$$

Если же  $S_n = 0$ , то (3) выполняется очевидным образом. Далее при  $1 \leq k \leq n$

$$|S_k| \leq \min \{|S_n| + |S_k|, |S_n| + |S_n - S_k|\} = |S_n| + Q_n,$$

так что

$$T_n \leq |S_n| + Q_n. \quad (4)$$

Из (3) и (4) следует

$$T_n \leq 3Q_n + \max_{1 \leq i \leq n} |\xi_i|. \quad (5)$$

Если мы покажем, что

$$Q_n \leq Q_n(m) + m \max_{1 \leq i \leq n} |\xi_i|. \quad (6)$$

то из (5) и (6) будет следовать утверждение леммы. При  $k \leq m$   $|S_k| \leq m \max_{1 \leq i \leq n} |\xi_i|$ , и очевидно, что (5) выполняется. Пусть  $k > m$ . Тогда

$$|S_k| \leq |S_{k-m}| + m \max_{1 \leq i \leq n} |\xi_i|, \quad Q_n \leq \max_{m < k \leq n} \min \left\{ |S_{k-m}| + m \max_{1 \leq i \leq n} |\xi_i|, |S_n - S_k| \right\},$$

откуда следует (5). Лемма доказана.

Пусть  $E_l = \{|S_i| \leq x, 1 \leq i < l, |S_l| > x\}$ ,  $l = 1, 2, \dots$ . Тогда  $E_i E_j = \emptyset$ ,  $i \neq j$ ,  $\bigcup_{l=1}^k E_l = \{T_k > x\}$ .

**Лемма 2.**

$$\mathbb{P}\{T_n \geq 3x + y\} \leq \mathbb{P}\left\{\bigcup_{l=1}^{n-m} (E_l, \max_{l+m \leq k \leq n} |S_n - S_k| > x)\right\} + \mathbb{P}\{(3m+1) \max_{1 \leq i \leq n} |\xi_i| > y\}.$$

Доказательство. Из (2) следует

$$\mathbb{P}\{T_n \geq 3x + y\} \leq \mathbb{P}\{Q_n(m) > x\} + \mathbb{P}\{(3m+1) \max_{1 \leq i \leq n} |\xi_i| > y\}. \quad (7)$$

Далее

$$\begin{aligned} \{Q_n(m) > x\} &\subseteq \left\{ \max_{m \leq k \leq n} \min \{T_{k-m}, |S_n - S_k|\} > x \right\} = \\ &= \left\{ \bigcup_{k=m}^n (T_{k-m} > x, |S_n - S_k| > x) \right\} = \left\{ \bigcup_{k=m}^n \bigcup_{l=1}^{k-m} (E_l, |S_n - S_k| > x) \right\} = \\ &= \left\{ \bigcup_{l=1}^{n-m} E_l \bigcup_{k=l+m}^n (|S_n - S_k| > x) \right\} = \left\{ \bigcup_{l=1}^{n-m} (E_l, \max_{l+m \leq k \leq n} |S_n - S_k| > x) \right\}. \end{aligned}$$

Последнее соотношение вместе с (7) даёт утверждение леммы.

Пусть теперь  $\{\xi_n\}$  — стационарная в узком смысле последовательность, удовлетворяющая условию  $\varphi$ -перемешивания.

Так как  $E_l \in \mathcal{F}_{\leq l}$ ,  $\left\{ \max_{l+m \leq k \leq n} |S_n - S_k| > x \right\} \in \mathcal{F}_{\geq l+m}$ , то

$$\begin{aligned} \mathbb{P}\left\{\bigcup_{l=1}^{n-m} (E_l, \max_{l+m \leq k \leq n} |S_n - S_k| > x)\right\} &= \sum_{l=1}^{n-m} \mathbb{P}\left\{E_l, \max_{l+m \leq k \leq n} |S_n - S_k| > x\right\} \leq \\ &\leq \sum_{l=1}^{n-m} \mathbb{P}(E_l) \left( \mathbb{P}\left\{\max_{l+m \leq k \leq n} |S_n - S_k| > x\right\} + \varphi(m) \right) \leq \\ &\leq \mathbb{P}\{T_n > x\} \left( \mathbb{P}\left\{\max_{0 \leq k \leq n} |S_n - S_k| > x\right\} + \varphi(m) \right) = \\ &= \mathbb{P}\{T_n > x\} (\mathbb{P}\{T_n > x\} + \varphi(m)). \end{aligned} \quad (8)$$

Величину  $\mathbb{P}\{T_n > a\} + \varphi(m)$  можно сделать сколь угодно малой выбором  $a$  и  $m$ , так что если  $\mathbb{P}\{T_n > a\} + \varphi(m) \leq \gamma_1 < 1$ , то из леммы 2 и (8) при  $x \geq a$  следует

$$\mathbb{P}\{T_n \geq 3x + y\} \leq \gamma_1 \mathbb{P}\{T_n > x\} + \mathbb{P}\left\{\max_{1 \leq i \leq n} |\xi_i| > \frac{y}{3m+1}\right\}.$$

Полученное неравенство является очевидным аналогом неравенства М. Пелиград. Чтобы ещё больше сблизить формулировки этих двух неравенств, можно заметить, что в условиях теоремы 1  $\mathbb{P}\{T_n > x\} \geq C\mathbb{P}\{|S_n| \geq \alpha x\}$ ,  $C > 0$ ,  $\alpha > 0$  (см., например, соотношения (3.5) и (3.9) в [1]).

Будем говорить, что стационарная последовательность  $\{\xi_n\}$  удовлетворяет условию  $\lambda$ -перемешивания, если существует функция  $\lambda(x) > 0$ ,  $\lambda(x) \downarrow 0$ ,  $x \downarrow 0$  такая, что

$$\sup \left\{ \frac{\mathbb{P}(AB)}{\mathbb{P}(A)\lambda(\mathbb{P}(B))} : A \in \mathcal{F}_{\leq 0}, B \in \mathcal{F}_{\geq l}, \mathbb{P}(A)\mathbb{P}(B) > 0 \right\} \leq 1$$

(см. [5]). Если последовательность  $\{\xi_n\}$  удовлетворяет условию  $\lambda$ -перемешивания, то

$$\begin{aligned} \mathbb{P} \left\{ \bigcup_{l=1}^{n-1} (E_l, \max_{l+1 \leq k \leq n} |S_n - S_k| > x) \right\} &= \sum_{l=1}^{n-1} \mathbb{P} \left\{ E_l, \max_{l+1 \leq k \leq n} |S_n - S_k| > x \right\} \leq \\ &\leq \sum_{l=1}^{n-1} \mathbb{P}(E_l) \lambda \left( \mathbb{P} \left\{ \max_{l+1 \leq k \leq n} |S_n - S_k| > x \right\} \right) \leq \mathbb{P}\{T_n > x\} \lambda(\mathbb{P}\{T_n > x\}). \end{aligned}$$

Величину  $\lambda(\mathbb{P}\{T_n > a\})$  можно сделать сколь угодно малым выбором  $a$ , так что если  $\lambda(\mathbb{P}\{T_n > a\}) \leq \gamma_2 < 1$ , то из леммы 2 с  $m = 1$  получаем следующий аналог неравенства М. Пелиград:

$$\mathbb{P}\{T_n \geq 3x + y\} \leq \gamma_2 \mathbb{P}\{T_n > x\} + \mathbb{P} \left\{ \max_{1 \leq i \leq n} |\xi_i| > \frac{y}{4} \right\}, \quad x \geq a.$$

Обозначим через  $L_{\leq n}$  и  $L_{\geq n}$  множества случайных величин с конечным вторым моментом и измеримых, соответственно, относительно  $\mathcal{F}_{\leq n}$  и  $\mathcal{F}_{\geq n}$ .

Будем говорить, что стационарная последовательность  $\{\xi_n\}$  удовлетворяет условию  $\rho$ -перемешивания, если

$$\rho(n) = \sup \left\{ \frac{|\mathbb{E} \xi \eta - \mathbb{E} \xi \mathbb{E} \eta|}{\sqrt{\mathbb{E} \xi^2 \mathbb{E} \eta^2}} : \xi \in L_{\leq 0}, \eta \in L_{\geq n} \right\} \rightarrow 0, \quad n \rightarrow \infty$$

(см., например, [7]). Взяв в этом определении  $\xi = \mathbf{1}_A$ ,  $A \in \mathcal{F}_{\leq 0}$ ,  $\eta = \mathbf{1}_B$ ,  $B \in \mathcal{F}_{\geq n}$ , получим

$$|\mathbb{P}\{AB\} - \mathbb{P}\{A\}\mathbb{P}\{B\}| \leq \rho(n) \sqrt{\mathbb{P}\{A\}\mathbb{P}\{B\}}. \quad (9)$$

**Теорема 2.** Пусть  $\{\xi_n\}$  стационарная последовательность и пусть при некоторых натуральных  $n$  и  $m$  таких, что  $n/r \geq 2$  и  $a > 0$  выполняется

$$2 \max_{2r \leq i \leq n} \mathbb{P}\{|S_i| > a\} + \sqrt{2n/r} \rho(r) \leq \gamma_3 < 1.$$

Тогда при любых  $x \geq 5a$

$$\max_{2r \leq i \leq n} \mathbb{P}\{|S_i| > x\} \leq \gamma_3 \mathbb{P}\{T_n > x/5\} + 2(n/r) \mathbb{P}\{\max_{1 \leq i \leq 2r} |S_i| > x/5\}.$$

Этот вариант своего неравенства М. Пелиград получила в [8]. Основное отличие этого неравенства от предыдущих в том, что в отличие от  $m$ , не зависящего от  $n$ , здесь  $r = r(n) \rightarrow \infty$  и используется более слабое условие перемешивания

Покажем, как предложенный выше подход позволяет получать результаты типа теоремы 2.

В лемму 2 вместо  $\xi_j$  подставим  $\eta_j = \sum_{i=1}^r \xi_{(j-1)r+i}$ ,  $j = 1, \dots, p = [n/r]$  (здесь  $[x]$  — целая часть  $x$ ), вместо  $S_k$  и  $T_k$  — соответственно  $U_k = \sum_{j=1}^k \eta_j$  и  $V_k = \max_{1 \leq j \leq k} |U_j|$ .

Пусть  $E_l = \{|U_j| \leq x, 1 \leq j < l, |U_l| > x\}$ ,  $l = 1, 2, \dots$ . Тогда  $E_i E_j = \emptyset$ ,  $i \neq j$ ,  $\bigcup_{l=1}^k E_l = \{V_k > x\}$ . Лемма 2 при  $m = 1$ ,  $x = y$  теперь будет выглядеть так:

$$\mathbb{P}\{V_p \geq 4x\} \leq \mathbb{P}\left\{\bigcup_{l=1}^{p-1} (E_l, \max_{l+1 \leq k \leq p} |U_n - U_k| > x)\right\} + \mathbb{P}\left\{\max_{1 \leq j \leq p} |\eta_j| > x/4\right\}. \quad (10)$$

Пусть  $\{\xi_n\}$  — стационарная последовательность. Имеем

$$\mathbb{P}\left\{\max_{1 \leq j \leq p} |\eta_j| > x/4\right\} \leq p\mathbb{P}\{|\eta_1| > x/4\} \leq (n/r)\mathbb{P}\{|S_r| > x/4\}. \quad (11)$$

Далее

$$\begin{aligned} p \left\{\bigcup_{l=1}^{p-1} (E_l, \max_{l+1 \leq k \leq p} |U_n - U_k| > x)\right\} &\leq \sum_{l=1}^{p-1} \left(\mathbb{P}\{E_l\} \mathbb{P}\left\{\max_{l+1 \leq k \leq p} |U_n - U_k| > x\right\}\right) + \\ &+ \rho(r) \sqrt{\mathbb{P}\{E_l\} \mathbb{P}\left\{\max_{l+1 \leq k \leq p} |U_n - U_k| > x\right\}} \leq \\ &\leq \mathbb{P}\{V_p > x\} \left(\mathbb{P}\{V_p > x\} + \rho(r) \sqrt{\mathbb{P}\{V_p > x\}} \sum_{l=1}^p \sqrt{\mathbb{P}\{E_l\}}\right). \end{aligned} \quad (12)$$

Нетрудно видеть, что

$$\sum_{l=1}^p \sqrt{\mathbb{P}\{E_l\}} \leq \sqrt{p \sum_{l=1}^p \mathbb{P}\{E_l\}} = \sqrt{p\mathbb{P}\{V_p > x\}}. \quad (13)$$

Из соотношений (10) – (13) следует

$$\mathbb{P}\{V_p > 4x\} \leq \mathbb{P}\{V_p > x\} \left(\mathbb{P}\{V_p > x\} + \sqrt{n/r} \rho(r)\right) + (n/r)\mathbb{P}\{|S_r| > x/4\}. \quad (14)$$

Так как  $\mathbb{P}\{V_p > x\} \leq \mathbb{P}\{T_n > x\}$ ,  $\mathbb{P}\{|S_{np}| > 4x\} \leq \mathbb{P}\{V_p > 4x\}$ , то из (14) следует

$$\begin{aligned} \mathbb{P}\{|S_n| > 5x\} &\leq \mathbb{P}\{|S_{np}| > 4x\} + \mathbb{P}\{|S_n - S_{np}| > x\} \leq \\ &\leq \gamma_4 \mathbb{P}\{T_n > x\} + (n/r + 1) \max_{1 \leq i \leq r} \mathbb{P}\{|S_i| > x/4\}, \quad \mathbb{P}\{T_n > x\} + \sqrt{n/r} \rho(r) \leq \gamma_4 < 1, \end{aligned}$$

что, очевидно, является аналогом теоремы 2.

## ЛИТЕРАТУРА

1. Peligrad M. An invariance principle for  $\varphi$ -mixing sequences // Ann. Probab. 1985. V. 13, N. 4. P. 1304–1313.
2. Peligrad M. On Ibragimov-Iosifescu conjecture for  $\varphi$ -mixing sequences // Stochastic Processis and their Application. 1990. V. 35. P. 293–308.
3. Гринь А.Г. Нормирующие последовательности в предельных теоремах для слабо зависимых величин // Теория вероят. и её примен. 1991. Т. 36, № 2. С. 285–300.
4. Гринь А.Г. Об областях притяжения для сумм зависимых величин // Теория вероятн. и её примен. 1990. Т. 35, № 2. С. 255–270.
5. Гринь А.Г. Области притяжения для последовательностей с перемешиванием // Сибирский математический журнал. 1990. Т. 31, № 1. С. 53–63.
6. Биллингсли П. Сходимость вероятностных мер. М. : Наука, 1977. 351 с.
7. Ибрагимов И.А., Линник Ю.В. Независимые и стационарно связанные величины. М. : Наука, 1965. 524 с.
8. Peligrad M. The convergence of moments in the central limit theorem for  $\rho$ -mixing sequences of random variables // Proceeding of the AMS. 1987. V. 101, N. 1. P. 142–148.

## РАЗЛИЧАЮЩИЕ И ГЛОБАЛЬНО ГИПЕРБОЛИЧЕСКИЕ ПРОСТРАНСТВА

**А.Н. Романов**

В работе изучаются условия, которые приводят хронологическое качество пространства-времени к достаточно сильному условию — глобальной гиперболичности. Ключевым моментом является поведение лоренцевой функции расстояния.

Работа посвящена исследованию поведения лоренцева расстояния на различающих и глобально гиперболических пространствах.

Будем использовать следующие вспомогательные результаты (см. 1-3).

**Лемма 1.** *Пространство-время  $(M, g)$  глобально гиперболично тогда и только тогда, когда оно сильно причинно и  $(M, g')$  удовлетворяет условию конечности расстояния для всех  $g' \in C(M, g)$ .*

Здесь через  $C(M, g)$  обозначен класс лоренцевых метрик на многообразии  $M$ , глобально конформных метрике  $g$  :

$$g' \in C(M, g) \Leftrightarrow g' = \Omega g$$

для некоторой гладкой функции  $\Omega : M \rightarrow (0, \infty)$ .

**Лемма 2.** *Пусть пространство-время  $(M, g)$  принадлежит классу  $A$ . Если для некоторых точек  $p, s \in M$ , множество  $J_p^+ \cap J_s^-$  не замкнуто в  $M$ , а  $I_p^+ \cap I_s^- \neq \emptyset$ , то тогда (замкнутое) множество  $cl(J_p^+ \cap J_s^-)$  не является компактным.*

Будем считать, что к классу пространств  $A$  не относятся лишь те пространства, в которых одновременно имеют место как явление захвата причинных кривых, так и явление конечной недостижимости (между некоторыми точками).

**Лемма 3.** *Пусть  $(M, g)$  — пространство-время. Если для некоторых точек  $p, s \in M$ , множество  $cl(J_p^+ \cap J_s^-)$  не является компактным, то существует лоренцева метрика  $g' \in C(M, g)$ , глобально конформная метрике  $g$  такая, что  $d_{g'}(a, b) = \infty$  для некоторых точек  $a, b \in M$ .*

Теперь применим указанные результаты к исследованию причинной структуры пространства-времени  $(M, g)$ , для которого условие конечности расстояния является инвариантом при конформных преобразованиях метрики  $g$ .

**Теорема 1.** Пусть  $(M, g)$  — различающее пространство-время класса  $A$ . Если пространство-время  $(M, g')$  удовлетворяет условию конечности расстояния для всех  $g' \in C(M, g)$ , то пространство-время  $(M, g)$  является глобально гиперболическим.

**Доказательство.** Доказательство этого утверждения сводится к следующим рассуждениям. Покажем сначала, что в условиях теоремы пространство-время  $(M, g)$  является причинно простым (то есть различающим с дополнительным условием, что множества  $J_p^+$  и  $J_p^-$  замкнуты для всех  $p \in M$ ).

Так как различаемость предполагается выполненной, остаётся доказать замкнутость множеств  $J_p^+$  и  $J_p^-$ . Проведём доказательство, например, для множества  $J_p^+$ : покажем, что оно замкнуто для любой точки  $p \in M$  (замкнутость  $J_p^-$  доказывается аналогично).

Допустим обратное: множество  $J_p^+$  не замкнуто, то есть существует точка  $q \in cl(J_p^+) \setminus J_p^+$ . Возьмём в  $I_q^+$  произвольную точку  $r$  и покажем, что множество  $J_p^+ \cap J_r^-$  не пусто и не является замкнутым..

Действительно, так как  $q \in cl(J_p^+)$ , то существует последовательность точек  $\{q_n\} \subset J_p^+$ , сходящаяся к  $q$ :  $q_n \rightarrow q$  (сходимость в исходной топологии многообразия  $M$ ). Так как  $q \in I_r^-$  (утверждение, эквивалентное тому, что  $r \in I_q^+$ ), а множество  $I_r^-$  открыто (см. [6], лемма 2.5), то некоторая окрестность точки  $q$  так же принадлежит  $I_r^-$ . Так как  $q_n \rightarrow q$ , то, начиная с некоторого  $n$ , все точки  $q_n$  попадают в эту окрестность, а следовательно и в  $I_r^-$ , то есть для достаточно больших  $n$  имеем:  $q_n \in I_r^-$  или иначе:  $q_n \ll r$ .

Таким образом,  $p \leq q_n$ ,  $q_n \ll r$ . Отсюда получаем:  $p \ll r$  (по свойствам соотношений  $\leq, \ll$ ), то есть  $r \in I_p^+$ .

В результате имеем: множество  $I_p^+ \cap I_r^-$  не пусто (так как существует времениподобная кривая  $\alpha_{pr}$ , соединяющая  $p$  с  $r$  — её существование гарантировано соотношением  $p \ll r$ ; и если  $s \in \alpha_{pr}$ ,  $s \neq p$ ,  $s \neq r$ , то  $s \in I_p^+ \cap I_r^-$ ).

Далее из соотношений

$$I_p^+ \subset J_p^+, I_r^- \subset J_r^-, I_p^+ \cap I_r^- \neq \emptyset$$

получаем:

$$J_p^+ \cap J_r^- \neq \emptyset.$$

Из приведённых ниже соотношений видно, что множество  $J_p^+ \cap J_r^-$  не замкнуто в  $M$ :

$$\begin{aligned} q \in I_r^- \subset int(J_r^-), q \in cl(J_p^+), q \notin J_p^+ &\implies \\ \implies q \in cl(J_p^+ \cap J_r^-) \setminus (J_p^+ \cap J_r^-) \end{aligned}$$

Тогда по лемме 2 получаем, что множество  $cl(J_p^+ \cap J_r^-)$  не компактно. Далее, следуя лемме 3, можно найти метрику  $g' \in C(M, g)$ , глобально конформную

---

метрике  $g$  такую, что пространство-время  $(M, g')$  не удовлетворяет условию конечности лоренцева расстояния, что противоречит условию теоремы.

Таким образом, множества  $J_p^+$  и  $J_p^-$  замкнуты для любой точки  $p \in M$ , что вместе с различаемостью  $(M, g)$  указывает на его причинную простоту, то есть пространство-время  $(M, g)$  является причинно простым.

Причинная простота пространства-времени  $(M, g)$  автоматически влечёт за собой его сильную причинность. Теперь по лемме 1 сразу получаем желаемый результат: пространство-время  $(M, g)$  является глобально гиперболическим. ■

## ЛИТЕРАТУРА

1. Бим Дж., Эрлих П. Глобальная лоренцева геометрия. М. : Мир, 1985.
2. Пенроуз Р. Структура пространства-времени. М. : Мир, 1972.
3. Malament D.B. The class of continuous timelike curves determines the topology of spacetime // J. Math. Phys. 1977. V. 18, N. 7. P. 1399–1404.

## **SURFACE CRITICALITY IN RANDOM FIELD SYSTEMS WITH CONTINUOUS SYMMETRY**

**A.A. Fedorenko**

We study the surface scaling behavior of a  $d$ -dimensional random field system with continuous  $O(N)$  symmetry. The system undergoes a paramagnetic-ferromagnetic transition above the lower critical dimension  $d_{lc} = 4$  for  $N > N_c = 2.835$ . Below the lower critical dimension and for  $N < N_c$  the system exhibits a quasi-long-range order with zero order parameter and a power-law decay of correlations. Using functional renormalization group we obtain the surface scaling laws describing the ordinary surface transition for  $d > d_{lc}$  and the behavior of correlations near the surface in the quasi-long-range ordered phase for  $d < d_{lc}$ .

### **1. Introduction**

Understanding of the critical properties of disordered systems attracted growing interest for decades. Among the most challenging problems is the critical behavior of the so-called random field systems in which the order parameter is linearly coupled to a random symmetry breaking field [19]. The effect of the random field (RF) disorder being more profound than many other types of disorder is much less understood. The prominent example is the random field Ising model (RFIM) whose complete understanding is still lacking despite significant numerical, analytical and experimental efforts [26]. The considerable progress has been achieved in recent years for the  $O(N)$  symmetric random field models. These models are relevant for diverse physical applications including amorphous magnets [18], liquid crystals in porous media [3, 12], nematic elastomers [17], critical fluids in aerogels [7, 9, 24], vortices in type II superconductors [2], and stochastic inflation in cosmology [20]. It was found that the expansion around the lower critical dimension of the the RF  $O(N)$  model  $d_{lc} = 4$  generates an infinite number of relevant operators whose flow can be studied using functional renormalization group [13, 15, 22, 31, 32]. Another challenging issue is the phase diagram of the RF systems below  $d_{lc}$ . It is known that for the RF model true long-range order is forbidden below  $d_{lc} = 4$  [6]. Nevertheless, quasi-long-range order (QLRO) with zero order parameter and an

infinite correlation length can persist even for  $d_{lc}^*(N) < d < d_{lc}$ , where  $d_{lc}^*(N)$  is the lower critical dimension for the paramagnetic-QLRO transition.

In general, the presence of boundaries in real systems can modify the behavior in the boundary region extended in the bulk only over distances of the order of the bulk correlation length. However, at the bulk critical point or in the QLRO phase, the bulk correlation length is infinite so that one can expect that the effect of boundaries is to be more pronounced. Indeed, the presence of the boundaries introduces a whole set of critical exponents describing the scaling behavior at and close to the boundary at criticality [1]. Several different classes of the surface transitions are known depending upon boundary conditions [23]. The different types of surface transitions have been studied for various systems with discrete and continuous symmetries using different methods, such as RG and numerical simulations [4, 28, 33]. However, not so much is known about the surface criticality in systems with RF disorder. The phase diagram of the 3D semi-infinite RFIM as a function of the ratio of bulk and surface interactions and the ratio of bulk and surface fields has been studied using a mean field approximation in [29]. The surface criticality of the RFIM has been studied numerically in [21]. It was also shown that the RF disorder on the surface of a 3D spin system with continuous symmetry destroys the long-range order in the bulk, and, instead, a QLRO emerges [14]. In this work we address the question of how the RF disorder in the bulk affect the behavior of spin systems with continuous symmetry in vicinity of free surfaces. We will consider the ordinary surface transition of the RF systems for  $d > 4$  and the order parameter correlations in the QLRO phase near a free surface for  $d < 4$ .

## 2. Model

Let us consider a  $d$ -dimensional semi-infinite  $O(N)$  spin system whose configuration is given by the  $N$ -component classical vector field  $\mathbf{s}(\mathbf{r})$  satisfying the fixed-length constraint  $|\mathbf{s}(\mathbf{r})|^2 = 1$ . The position vector  $\mathbf{r} = (\mathbf{x}, z)$  has a  $(d - 1)$ -dimensional component  $\mathbf{x}$  parallel to the surface and a one-dimensional component  $z \geq 0$  that is perpendicular to the surface  $z = 0$ . It is convenient to introduce short notations for the volume integral over half space  $\int_V := \int_0^\infty dz \int d^{d-1}x$  and for the surface integral  $\int_S := \int d^{d-1}x$ . The large-scale behavior of the disordered spin system can be described by the effective Hamiltonian

$$\mathcal{H}[\mathbf{s}] = \mathcal{H}_0[\mathbf{s}] + \mathcal{H}_{\text{surf}}[\mathbf{s}] + \mathcal{H}_{\text{dis}}[\mathbf{s}], \quad (1)$$

consisting of the sum of three terms which result from the semi-infinite bulk, surface and disorder in the bulk. The contributions from the semi-infinite bulk and the surface can be written in its simplest form as [5]:

$$\mathcal{H}_0[\mathbf{s}] = \int_V \left[ \frac{1}{2} (\nabla \mathbf{s}(\mathbf{r}))^2 - \mathbf{h} \cdot \mathbf{s}(\mathbf{r}) \right], \quad \mathcal{H}_{\text{surf}}[\mathbf{s}] = - \int_S \mathbf{h}_1 \cdot \mathbf{s}(\mathbf{x}), \quad (2)$$

where for simplicity we assume that the surface magnetic field  $\mathbf{h}_1$  has the same direction as the bulk field  $\mathbf{h}$ . We consider a quite general type of bulk disorder

such that its potential can be expanded in spin variables as follows

$$\mathcal{H}_{\text{dis}}[\mathbf{s}] = - \int_V \sum_{\mu=1}^{\infty} \sum_{i_1 \dots i_{\mu}} h_{i_1 \dots i_{\mu}}^{(\mu)}(\mathbf{r}) s_{i_1}(\mathbf{r}) \dots s_{i_{\mu}}(\mathbf{r}). \quad (3)$$

The coefficients  $h_{i_1 \dots i_{\mu}}^{(\mu)}(\mathbf{r})$  are Gaussian random variables with zero mean and variances given by

$$\overline{h_{i_1 \dots i_{\mu}}^{(\mu)}(\mathbf{r}) h_{i_1' \dots i_{\mu}'}^{(\mu)}(\mathbf{r}') } = \delta^{\mu\nu} \delta_{i_1 j_1} \dots \delta_{i_{\mu} j_{\mu}} r_{\mu} \delta(\mathbf{r} - \mathbf{r}'). \quad (4)$$

The first two coefficients have simple physical interpretation:  $h_i^{(1)}$  is a random magnetic field and  $h_{ij}^{(2)}$  is a second-rank random anisotropy. The higher order coefficients  $h^{(\mu)}$  are higher order random anisotropies. As was shown in [15], even if the system has only finite number of nonzero bare  $h^{(\mu)}$ , the RG transformations will generate an infinite set of higher-order anisotropies.

To average over disorder we use the replica trick and introduce  $n$  replicas of the original system. Averaging their joint partition function over disorder we obtain the replicated Hamiltonian as

$$\mathcal{H}_n = \int_V \left\{ \sum_{a=1}^n \left[ \frac{1}{2} (\nabla \mathbf{s}_a(\mathbf{r}))^2 - \mathbf{h} \cdot \mathbf{s}_a(\mathbf{r}) \right] - \frac{1}{2T} \sum_{a,b=1}^n \mathcal{R}(\mathbf{s}_a(\mathbf{r}) \cdot \mathbf{s}_b(\mathbf{r})) \right\} - \sum_{a=1}^n \int_S \mathbf{h}_1 \cdot \mathbf{s}_a(\mathbf{x}), \quad (5)$$

where we have defined the function  $\mathcal{R}(z) = \sum_{\mu} r_{\mu} z^{\mu}$ . The properties of the original disordered system (1) can be extracted in the limit  $n \rightarrow 0$ .

Power counting shows that  $d_{lc} = 4$  is the lower critical dimension of the model (5). Above the lower critical dimension the RF systems undergo a paramagnetic - ferromagnetic transition. The scaling behavior at criticality is controlled by a zero temperature fixed point (FP) similar to the RFIM, reflecting the fact that disorder dominates over the thermal fluctuations. However, the temperature is dangerously irrelevant. For instance, this results in violation of the usual hyperscaling relation and the appearance of a new universal exponent  $\theta$  that modifies the hyperscaling relation to [26]:

$$\nu(d - \theta) = 2 - \alpha, \quad (6)$$

where  $\nu$  and  $\alpha$  are the correlation length and the specific heat exponents. One also expects a dramatic slowing down as the transition is approached with the characteristic relaxation time  $\ln \tau \sim t_1^{-\nu\theta}$ , where  $t_1 = |T - T_c|/T_c$  is the reduced temperature [16]. The magnetization in the bulk and on the surface vanish at the transition according to

$$\sigma(t_1) \sim t_1^{\beta}, \quad \sigma_1(t_1) \sim t_1^{\beta_1}, \quad (7)$$

where we have introduced the bulk and the surface magnetization exponents. At the critical point  $t_1 = 0$  small magnetic field in the bulk  $\mathbf{h}$  and on the surface  $\mathbf{h}_1$  can induce the magnetization in the bulk and also on the surface according to

$$\sigma(h) \sim h^{1/\delta}, \quad \sigma_1(h) \sim h^{1/\delta_1}, \quad \sigma_1(h_1) \sim h_1^{1/\delta_{11}}, \quad (8)$$

where we define the exponents  $\delta$ ,  $\delta_1$  and  $\delta_{11}$ . Below the lower critical dimension  $d_{lc}$  a QLRO phase with zero magnetization can emerge. At criticality or in the QLRO phase, the correlation functions of the order parameter exhibit scaling behavior. Due to dangerous irrelevance of the temperature the connected and disconnected correlation functions scale with different exponents. We define the connected and disconnected correlation functions of the two local operators  $A$  and  $B$  as

$$\begin{aligned} [A(\mathbf{r}) \cdot B(\mathbf{r}')]_{\text{con}} &:= \overline{\langle A(\mathbf{r}) \cdot B(\mathbf{r}') \rangle - \langle A(\mathbf{r}) \rangle \cdot \langle B(\mathbf{r}') \rangle}, \\ [A(\mathbf{r}) \cdot B(\mathbf{r}')]_{\text{dis}} &:= \overline{\langle A(\mathbf{r}) \rangle \cdot \langle B(\mathbf{r}') \rangle - \langle A(\mathbf{r}) \rangle \cdot \langle B(\mathbf{r}') \rangle}. \end{aligned}$$

Here the angular brackets denote the thermal averaging and the overbar stands for the disorder averaging. For instance, the connected and disconnected correlation functions of spins in the bulk scale independently as

$$[\mathbf{s}(\mathbf{r}) \cdot \mathbf{s}(\mathbf{r}')]_{\text{con}} \sim \frac{1}{|\mathbf{r} - \mathbf{r}'|^{d-2+\eta}}, \quad [\mathbf{s}(\mathbf{r}) \cdot \mathbf{s}(\mathbf{r}')]_{\text{dis}} \sim \frac{1}{|\mathbf{r} - \mathbf{r}'|^{d-4+\bar{\eta}}}. \quad (9)$$

Following the general scaling picture of the surface critical phenomena we introduce the surface exponents  $\eta_{\perp}$  and  $\bar{\eta}_{\perp}$  which replace the bulk exponents  $\eta$  and  $\bar{\eta}$  in equations (9) when one of the points  $\mathbf{r}$  or  $\mathbf{r}'$  belongs to the surface:

$$[\mathbf{s}(\mathbf{x}, z) \cdot \mathbf{s}(\mathbf{x}', 0)]_{\text{con}} \sim \frac{1}{((\mathbf{x} - \mathbf{x}')^2 + z^2)^{(d-2+\eta_{\perp})/2}}, \quad (10)$$

$$[\mathbf{s}(\mathbf{x}, z) \cdot \mathbf{s}(\mathbf{x}', 0)]_{\text{dis}} \sim \frac{1}{((\mathbf{x} - \mathbf{x}')^2 + z^2)^{(d-4+\bar{\eta}_{\perp})/2}}. \quad (11)$$

We also define the surface exponents  $\eta_{\parallel}$  and  $\bar{\eta}_{\parallel}$  that describe the connected and disconnected correlation function when both points lie on the surface:

$$[\mathbf{s}(\mathbf{x}) \cdot \mathbf{s}(\mathbf{x}')]_{\text{con}} \sim \frac{1}{|\mathbf{x} - \mathbf{x}'|^{d-2+\eta_{\parallel}}}, \quad [\mathbf{s}(\mathbf{x}) \cdot \mathbf{s}(\mathbf{x}')]_{\text{dis}} \sim \frac{1}{|\mathbf{x} - \mathbf{x}'|^{d-4+\bar{\eta}_{\parallel}}}. \quad (12)$$

### 3. Functional renormalization group

In the limit of low temperature and weak disorder the configuration of the system is fluctuating around the completely ordered state in which all replicas of all spins align along the same direction which is parallel to  $\mathbf{h}$  and  $\mathbf{h}_1$ . It is convenient to split the order parameter  $\mathbf{s}_a = (\sigma_a, \boldsymbol{\pi}_a)$  into the  $(N - 1)$ -component vector  $\boldsymbol{\pi}_a$  which is perpendicular to this direction and the component  $\sigma_a = \sqrt{1 - \boldsymbol{\pi}_a^2}$  being

parallel to this direction. Then the effective action of the system can be written as

$$\begin{aligned} \mathcal{S}[\boldsymbol{\pi}] = & \frac{1}{T} \sum_{a=1}^n \left\{ \int_V \left[ \frac{1}{2} (\nabla \boldsymbol{\pi}_a)^2 + \frac{(\boldsymbol{\pi}_a \cdot \nabla \boldsymbol{\pi}_a)^2}{2(1 - \boldsymbol{\pi}_a^2)} - h \sigma_a \right] - \int_S h_1 \sigma_a \right\} - \\ & - \frac{1}{2T^2} \sum_{a,b=1}^n \int_V \mathcal{R}(\boldsymbol{\pi}_a \cdot \boldsymbol{\pi}_b + \sigma_a \sigma_b). \end{aligned} \quad (13)$$

In general one has to add to the action (13) the terms like  $\delta^d(0) \int_V \ln(1 - \boldsymbol{\pi}_a^2)$  generated by the Jacobian of the transformation from  $\mathbf{s}_a$  to  $\boldsymbol{\pi}_a$ . However, in what follows we will use the dimensional regularization scheme [5] in which  $\delta^d(0) = 0$  so we can ignore these terms in action (13) from the beginning.

Let us denote averaging with the action (13) by double angular brackets and introduce the following correlation functions

$$G_{\alpha,\beta}^{(L,K)}(\mathbf{r}, \mathbf{x}) = \left\langle \left\langle \prod_{\nu=1}^L \pi_{\alpha_\nu}(\mathbf{r}_\nu) \prod_{\mu=1}^K \pi_{\beta_\mu}(\mathbf{x}_\mu) \right\rangle \right\rangle, \quad (14)$$

where  $L$  points  $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_L)$  are off surface and  $K$  points  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_K)$  are sitting on the surface. In equation (14) we have used a short notation  $\alpha = (\alpha_1, \dots, \alpha_L)$  where each  $\alpha_\nu$  stands for the component number  $i_\nu$  and the replica number  $a_\nu$ . The similar holds for  $\beta$ . Using correlation functions (14) one can compute the connected and disconnected functions defined in equations (9). However, since we are interested only in the scaling behavior it is more convenient to consider the similar correlation functions not for  $\mathbf{s}$  but for  $\boldsymbol{\pi}$  fields.

Expanding the effective action (13) in small  $\boldsymbol{\pi}$  we will treat the quadratic part as a free action and the rest of the infinite series as interaction vertices. Then the correlation functions (14) can be expressed in terms of Feynman diagrams which give the low temperature and small disorder expansion. In practical calculations it is convenient to perform the Fourier transform with respect to  $\mathbf{x}$ :  $\hat{\boldsymbol{\pi}}(\mathbf{q}, z) = \int d^{d-1}x \boldsymbol{\pi}(\mathbf{x}, z) e^{-i\mathbf{q}\cdot\mathbf{x}}$  and define  $\int_q := \int d^{d-1}q / (2\pi)^{d-1}$ . The quadratic terms give the free propagator

$$\hat{G}_q^{(0)}(z, z') = \frac{1}{2\bar{q}} \left[ e^{-\bar{q}|z-z'|} + \frac{\bar{q} - h_1}{\bar{q} + h_1} e^{-\bar{q}(z+z')} \right], \quad (15)$$

where we have introduced the shorthand notation  $\bar{q} := (q^2 + h)^{1/2}$ . The free surface corresponds to the limit  $h_1 \rightarrow 0$  in which equation (15) becomes the Neumann propagator consisting of the bulk part and the image part. In what follows we will use the Neumann propagator as the bare one and treat the terms proportional to  $h_1$  as soft insertions [5, 11].

The correlation functions (14) calculated perturbatively in small disorder and temperature suffer from the ultraviolet divergences. To avoid mixture with infrared singularities in the  $O(N)$ -noninvariant correlation functions it is convenient to keep  $\mathbf{h} \neq 0$ . The ultraviolet divergences can be converted into poles in  $\varepsilon = d - 4$  using dimensional regularization. To renormalize the theory one has to absorb these

poles into finite number of  $Z$ -factors. However, all the Taylor coefficients  $r_\mu$  of the disorder correlator  $\mathcal{R}(\phi)$  turn out to be relevant operators so that one has to introduce renormalization of the whole function. Since the scaling behavior is controlled by a zero temperature FP we will disregard all terms involving more than two replicas which are suppressed in the limit  $T \rightarrow 0$ . The renormalization of the disorder simplifies by changing variables:  $\mathcal{R}(\phi) = R(z)$  where  $z = \cos \phi$ , for instance,  $\mathcal{R}'(1) = -R''(0)$ . In terms of the variable  $\phi$ , the function  $R(\phi)$  becomes periodic with period  $2\pi$  in the RF case. The relation between the renormalized and the bare correlation functions reads

$$G^{(L,K)}(\mathbf{r}; T, h, h_1, R, \mu) = Z_\pi^{-(L+K)/2} Z_1^{-K/2} \mathring{G}^{(L,K)}(\mathbf{r}; \mathring{T}, \mathring{h}, \mathring{h}_1, \mathring{R}). \quad (16)$$

where circles denote the bare quantities and  $\mu$  is an arbitrary momentum scale. The ultraviolet divergences are absorbed into  $Z$ -factors according to

$$\mathring{\pi} = Z_\pi^{1/2} \pi, \quad \mathring{\pi}|_s = (Z_\pi Z_1)^{1/2} \pi|_s, \quad \mathring{h} = \mu^2 Z_T Z_\pi^{-1/2} h, \quad (17)$$

$$\mathring{h}_1 = \mu Z_T (Z_\pi Z_1)^{-1/2} h_1, \quad \mathring{T} = \mu^{2-d} Z_T T, \quad \mathring{R} = \mu^{4-d} K_d^{-1} Z_R[R], \quad (18)$$

where  $(2\pi)^d K_d = 2\pi^{d/2}/\Gamma(d/2)$  is the surface area of a  $d$ -dimensional unit sphere and  $\Gamma(x)$  is the Euler gamma function. In equation (18)  $Z_R[R]$  is a functional acting on the renormalized disorder correlator  $R(\phi)$  which has the following loop expansion:

$$Z_R[R] = R + \delta^{(1)}(R, R) + \delta^{(2)}(R, R, R) + \dots, \quad (19)$$

where  $\delta^{(1)}(R, R)$  is bilinear in  $R$  and proportional to  $1/\varepsilon$ , while  $\delta^{(2)}(R, R, R)$  is cubic in  $R$  and contains terms of order  $1/\varepsilon$  and  $1/\varepsilon^2$ . According to equations (17) the surface field  $\pi|_s$  renormalizes differently from the field  $\pi$  in the bulk. The new factor  $Z_1$  serves to cancel the additional ultraviolet divergences in Feynman diagrams arising from the image part of the Neumann propagator  $\hat{G}_q^{(0)}(z, z')$  for  $z' \rightarrow 0$ . The renormalized theory is not unique and depends on the scale  $\mu$ . Using this fact we will derive the functional renormalization group equation.

We now consider how the scaling behavior can be extracted from the renormalized theory. Using the independence of the bare theory on the momentum scale  $\mu$  one can derive the flow equations for the renormalized correlation functions differentiating both sides of equation (16) with respect to  $\mu$  at fixed bare quantities. One finds that the renormalized correlation functions satisfy the following FRG equation

$$\left[ \mu \partial_\mu + (d-2-\zeta_T) T \partial_T - \zeta_h h \partial_h - \zeta_{h_1} h_1 \partial_{h_1} + \frac{L}{2} \zeta_\pi + \frac{K}{2} (\zeta_\pi + \zeta_1) - \int d\phi \beta[R(\phi)] \frac{\delta}{\delta R(\phi)} \right] G^{(L,K)} = 0, \quad (20)$$

where the integral in the last line is taken over a period, i.e.,  $(0, \pi)$  for RA and  $(0, 2\pi)$  for RF models and we have introduced the scaling functions:

$$\zeta_i = \mu \partial_\mu \ln Z_i|_0, \quad (i = T, \pi, 1), \quad (21)$$

$$\zeta_h = 2 + \zeta_T - \zeta_\pi/2, \quad \zeta_{h_1} = 1 + \zeta_T - (\zeta_\pi + \zeta_1)/2, \quad (22)$$

$$\beta[R] = -\mu \partial_\mu R(\phi)|_0. \quad (23)$$

Here the zero indicates that the derivatives are taken at fixed bare quantities. Flow equations similar to equation (20) hold also for the correlation functions in which some or all the fields  $\pi_a(\mathbf{r})$  are replaced by  $\sigma_a(\mathbf{r})$  and for other observables, e.g., the correlation length and the magnetization.

The long-distance physics can be obtained from the solution of the FRG equation (20) in the limit of  $\mu \rightarrow 0$ . The renormalized disorder correlator and the temperature flow according to

$$-\mu\partial_\mu R(\phi) = \beta[R], \quad (24)$$

$$-\mu\partial_\mu \ln T = 2 - d + \zeta_T. \quad (25)$$

The scaling behavior is controlled by a zero temperature FP  $\beta[R^*] = 0$  with  $R^*$  of order  $\varepsilon$  and  $T^* = 0$ . Indeed, according to equation (25), the temperature is irrelevant, i.e. it flows to 0 in the limit  $\mu \rightarrow 0$  for  $d > 2$  and for sufficiently small  $\zeta_T = O(R)$ . Although one expects that  $\zeta_T$  is small in the vicinity of the FP, one has to take caution whether the zero temperature FP survives in three dimensions where  $\zeta_T \sim \varepsilon$  is negative [13]. The stability of the FP can be checked by computing the eigenvalues of the disorder flow equation (24) linearized about the FP solution:  $R(\phi) = R^*(\phi) + \sum_i t_i \Psi_i(\phi)$ . Since one expects that for  $d > 4$  ( $\varepsilon > 0$ ) the FP  $R^*(\phi)$  describes the paramagnetic-ferromagnetic transition it has to be unstable in a single direction  $\Psi_1(\phi)$  with eigenvalue  $\lambda_1 > 0$ :  $\beta[R^* + t_1 \Psi_1] = \lambda_1 t_1 \Psi_1 + O(t_1^2)$ . In the vicinity of the zero temperature FP that controls the paramagnetic-ferromagnetic transition, the FRG equation for the correlation length  $\xi$  can be written as

$$\left[ \mu\partial_\mu - \lambda_1 t_1 \frac{\partial}{\partial t_1} \right] \xi(\mu, t_1) = 0. \quad (26)$$

Dimensional analysis implies that  $\xi(\mu, t_1) = \mu^{-1} \bar{\xi}(t_1)$ . This reduces equation (26) to an ordinary differential equation (ODE) whose solution is  $\xi \sim \mu^{-1} t_1^{-1/\lambda_1}$ . The latter describes divergence of the correlation length on the critical line at zero temperature when the strength of disorder approaches the critical value. Assuming that along the transition line at finite temperature  $t_1 \sim T - T_c$  we find that the positive eigenvalue  $\lambda_1$  gives the critical exponent of the correlation length  $\nu = 1/\lambda_1$ . For  $d < 4$  ( $\varepsilon < 0$ ) the FP becomes stable and describes a QLRO phase. The fluctuations exhibit power-law correlations in the whole QLRO phase so that the correlation length  $\xi$  is always infinite down to the lower critical dimension of the QLRO - paramagnetic transition.

Let us consider the solution of equation (20) for the connected two-point correlation functions. The dangerous irrelevance of the temperature manifests itself in the fact that the connected (bulk or surface) two point functions are proportional to  $T$  in the low temperature limit. Thus, setting  $h = h_1 = 0$  and  $R = R^*$  we can rewrite equation (20) as

$$\left[ \mu\partial_\mu + \frac{1}{2}(L + K)\zeta_\pi^* + \frac{K}{2}\zeta_1^* + \theta \right] G_{\text{con}}^{(L,K)} = 0, \quad (27)$$

where the asterisk denotes that the function is computed at the FP. In equation (27) we have defined the exponent

$$\theta = d - 2 - \zeta_T^*, \quad (28)$$

which describes the flow of the temperature (25) in the vicinity of the FP and has been introduced ad hoc in the modified hyperscaling relation (6). Using the method of characteristics and dimensional analysis one can write the solution of equation (27) in the form

$$G_{\text{con}}^{(L,K)}(rb; R^*) = b^{-(\frac{1}{2}(L+K)\zeta_\pi^* + K\zeta_1^*/2 + \theta)} f_c(r; R^*). \quad (29)$$

Considering the connected two point functions (29) with  $(L = 2, K = 0)$ ,  $(L = 1, K = 1)$ , and  $(L = 0, K = 2)$  we derive the critical exponents:

$$\eta = \zeta_\pi^* - \zeta_T^*, \quad \eta_\perp = \zeta_\pi^* + \zeta_1^*/2 - \zeta_T^*, \quad \eta_\parallel = \zeta_\pi^* + \zeta_1^* - \zeta_T^*. \quad (30)$$

We next turn to the disconnected two-point correlation functions. At variance with the connected correlation functions they are not proportional to the temperature. Thus, at  $h = h_1 = T = 0$  they satisfy the same equation (27) but without the term  $\theta$  in large square brackets. The solution of the latter FRG equation is given by

$$G_{\text{dis}}^{(L,K)}(rb; R^*) = b^{-(\frac{1}{2}(L+K)\zeta_\pi^* + K\zeta_1^*/2)} f_d(r; R^*). \quad (31)$$

Repeating the analysis we did for the connected functions, we arrive at

$$\bar{\eta} = 4 - d + \zeta_\pi^* = 2 + \eta - \theta, \quad \bar{\eta}_\perp = 4 - d + \zeta_\pi^* + \zeta_1^*/2 = 2 + \eta_\perp - \theta, \quad (32)$$

$$\bar{\eta}_\parallel = 4 - d + \zeta_\pi^* + \zeta_1^* = 2 + \eta_\parallel - \theta. \quad (33)$$

Note that the exponents (30) and (32)–(33) are related by

$$2\eta_\perp = \eta + \eta_\parallel, \quad 2\bar{\eta}_\perp = \bar{\eta} + \bar{\eta}_\parallel. \quad (34)$$

Finally we study the profile of the spontaneous magnetization below and at the paramagnetic-ferromagnetic transition for  $d > d_{lc}$ . The magnetization as a function of the distance to the surface  $z$ , the reduced temperature  $t_1$ , and the bulk and surface magnetic fields  $h$  and  $h_1$  satisfies the following flow equation

$$\left[ \mu \partial_\mu - \zeta_h^* h \partial_h - \zeta_{h_1}^* h_1 \partial_{h_1} + \frac{1}{2} \zeta_\pi^* + \frac{j}{2} \zeta_1^* - \lambda_1 t_1 \frac{\partial}{\partial t_1} \right] \sigma(z, t_1, h, h_1) = 0. \quad (35)$$

Here  $j = 0$  and  $z > 0$  corresponds to the bulk magnetization  $\sigma$  while  $j = 1$  and  $z = 0$  gives the surface magnetization  $\sigma_1$ . The solution of equation (35) can be written as

$$\sigma(z, t_1, h, h_1) = b^{-(\frac{1}{2}\zeta_\pi^* + \frac{j}{2}\zeta_1^*)} \sigma(zb^{-1}, t_1 b^{\lambda_1}, hb^{\zeta_h^*}, h_1 b^{\zeta_{h_1}^*}). \quad (36)$$

We first consider the profile for  $h = h_1 = 0$ . The solution (36) interpolates between the surface magnetization  $\sigma_1(t_1) \sim t_1^{(\zeta_\pi^* + \zeta_1^*)/(2\lambda_1)}$  at  $z \approx 0$  and the bulk magnetization  $\sigma(t_1, z) \sim t_1^{\zeta_\pi^*/(2\lambda_1)}$  for  $z \gg \xi$ . Reexpressing the latter in terms of  $\nu, \bar{\eta}$ , and  $\bar{\eta}_\parallel$  we obtain that the bulk and the surface magnetization exponents defined in equation (7) are given by

$$\beta = \frac{1}{2}\nu(d - 4 + \bar{\eta}), \quad \beta_1 = \frac{1}{2}\nu(d - 4 + \bar{\eta}_\parallel). \quad (37)$$

At the critical point  $t_1 = 0$  and finite external fields we find that  $\sigma(h) \sim h^{\zeta_\pi^*/(2\zeta_h^*)}$  in the bulk and  $\sigma_1(h) \sim h^{(\zeta_\pi^* + \zeta_1^*)/(2\zeta_h^*)}$  or  $\sigma_1(h_1) \sim h_1^{(\zeta_\pi^* + \zeta_1^*)/(2\zeta_{h_1}^*)}$  at the surface. Thus, the exponents  $\delta$ ,  $\delta_1$ , and  $\delta_{11}$  defined in equations (8) satisfy the following scaling relations:

$$\frac{\delta - 1}{2 - \eta} = \frac{\nu}{\beta}, \quad \frac{\delta_1 - \beta/\beta_1}{2 - \eta} = \frac{\nu}{\beta_1}, \quad \frac{\delta_{11} - 1}{1 - \eta_\parallel} = \frac{\nu}{\beta_1}. \quad (38)$$

#### 4. Surface exponents to one-loop order

We now renormalize the both semi-infinite RF and RA models to one-loop order and explicitly calculate the surface critical exponents to first order in  $\varepsilon = d - 4$ . The factors  $Z_\pi$ ,  $Z_T$  and  $Z_R[R]$  defined in equations (17)–(19) are the same that appear in the case of the infinite systems. They have been calculated in several works up to two-loop order [13, 15, 22, 31]. To one-loop order they read

$$\begin{aligned} Z_\pi &= 1 - (N - 1) \frac{R''(0)}{\varepsilon} + O(R^2), \quad Z_T = 1 - (N - 2) \frac{R''(0)}{\varepsilon} + O(R^2), \quad (39) \\ \varepsilon \delta^{(1)}(R, R) &= \frac{1}{2} R''(\phi)^2 - R''(0) R''(\phi) - (N - 2) \left\{ R''(0) [2R(\phi) + \right. \\ &\quad \left. + R'(\phi) \cot \phi] - \frac{1}{2 \sin^2 \phi} [R'(\phi)]^2 \right\}. \quad (40) \end{aligned}$$

The new factor  $Z_1$  that eliminates the poles resulting from the presence of the surface can be determined from the renormalization of the two point function  $\mathring{G}^{(1,1)}(p, z; \mathring{h}, \mathring{T}, \mathring{R})$  which reads to one-loop order

$$\begin{aligned} \mathring{G}^{(1,1)}(p, z; \mathring{h}, \mathring{T}, \mathring{R}) &= \mathring{T} \frac{e^{-\bar{p}z}}{\bar{p}} \left\{ 1 - \frac{K_d}{4\varepsilon} \mathring{R}''(0) \times \right. \\ &\quad \left. \times \left[ (N - 3) \left( \frac{\mathring{h}}{\bar{p}^2} + \frac{z\mathring{h}}{\bar{p}} \right) + 2(N + 1) \right] + O(\mathring{R}^2) \right\}, \quad (41) \end{aligned}$$

where  $\bar{p} = (p^2 + \mathring{h}^2)^{1/2}$ . The factor  $Z_1$  can be found from the renormalization condition

$$Z_\pi^{-1} Z_1^{-1/2} \mathring{G}^{(1,1)}(p, z; \mathring{h}, \mathring{T}, \mathring{R}) = \text{finite for } \varepsilon \rightarrow 0, \quad (42)$$

where the bare  $\mathring{h}, \mathring{T}, \mathring{R}$  are replaced by the renormalized  $h, T$  and  $R$  according to equations (17)-(18). We obtain

$$Z_1 = 1 - (N - 1) \frac{R''(0)}{\varepsilon} + O(R^2). \quad (43)$$

Thus, to one loop order we have  $Z_1 = Z_\pi + O(R^2)$ . Using equations (21)-(23) we calculate the scaling functions

$$\zeta_T = -(N - 2)R''(0) + O(R^2), \quad \zeta_\pi = \zeta_1 = -(N - 1)R''(0) + O(R^2), \quad (44)$$

and the beta function

$$\begin{aligned} \beta[R] = & -\varepsilon R(\phi) + \frac{1}{2}R''(\phi)^2 - R''(0)R''(\phi) - (N - 2) \times \\ & \times \left\{ R''(0)[2R(\phi) + R'(\phi) \cot \phi] - \frac{1}{2\sin^2 \phi} [R'(\phi)]^2 \right\} + O(R^2) \end{aligned} \quad (45)$$

to one-loop order. The solution of the FP equation  $\beta[R^*] = 0$  with the beta function (45) has been analyzed for different values of  $N$  and different sign of  $\varepsilon$  in [13, 22, 31]. We first assume for granted that the flow has a FP  $R^*(\phi)$  which is a  $\pi$ -periodic function for the RA model and a  $2\pi$ -periodic function for the RF model. Then, the surface critical exponents can be computed to one loop using equations (30) and (32)-(33). They give

$$\eta = -R^{*''}(0), \quad \bar{\eta} = -\varepsilon - (N - 1)R^{*''}(0), \quad (46)$$

$$\eta_\perp = -\frac{N + 1}{2}R^{*''}(0), \quad \bar{\eta}_\perp = -\varepsilon - \frac{3}{2}(N - 1)R^{*''}(0), \quad (47)$$

$$\eta_\parallel = -NR^{*''}(0), \quad \bar{\eta}_\parallel = -\varepsilon - 2(N - 1)R^{*''}(0). \quad (48)$$

The other surface exponents are related to the exponents (46)-(48) by the scaling relations (37) and (38).

#### 4.1. Paramagnetic-ferromagnetic transition for $d > 4$ ( $\varepsilon > 0$ )

The RF model is described by  $R(\phi)$  which is a  $2\pi$ -periodic function. Numerical solution of the FP equation shows that for  $d > 4$  a  $2\pi$ -periodic solution exists only for  $N > N_c = 2.83474$ . It has  $R^{*''}(0) < 0$  and it disappears when  $N \rightarrow N_c^+$ . This cuspy FP is once unstable with the positive eigenvalue  $\lambda_1 = \varepsilon$ . Thus, the correlation length exponent  $\nu = 1/\varepsilon + (\varepsilon^0)$  coincides with the DR prediction to one-loop order. Remarkably, the non-zero  $R^{*''}(0^+)$  vanishes for  $N > N^* = 18 + O(\varepsilon)$ . The non-analyticity becomes weaker as  $N$  increases and starts with  $R^{*(2p(N)+1)}(0^+) \neq 0$  where  $p \sim N$  [22, 30, 31]. Weaker non-analyticity results in restoring the DR critical exponents for  $N > N^*$ . The critical exponents  $\eta_i$  and  $\bar{\eta}_i$  computed using equations (46)-(48) as functions of  $N$  are shown in the right panel of Figure 1. With increasing  $N$  they monotonically decay approaching the DR values at  $N = N^*$  and satisfying the inequalities:  $\eta < \bar{\eta} < \eta_\perp < \bar{\eta}_\perp < \eta_\parallel < \bar{\eta}_\parallel$ . The bulk and surface magnetization exponents  $\beta$  and  $\beta_1$  calculated for different  $N$

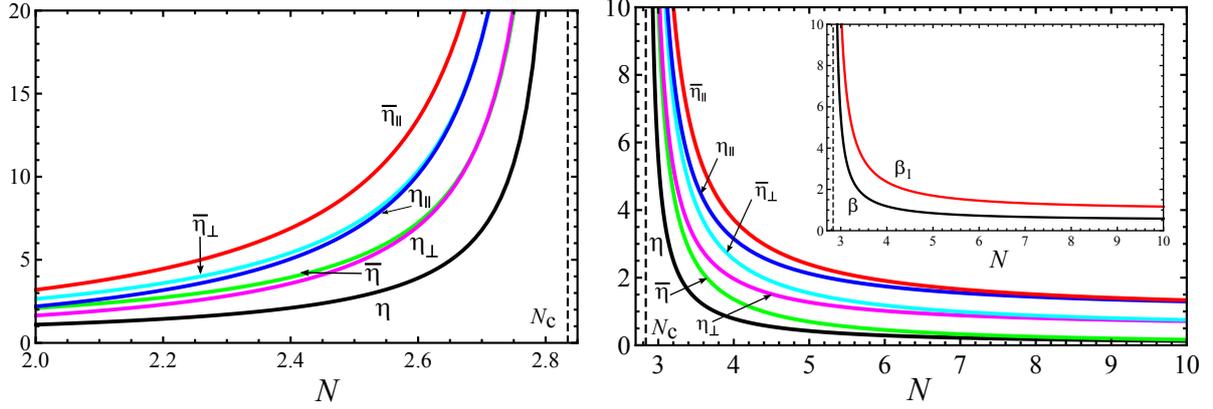


Figure 1. **Left panel:** the critical exponents  $\eta_i$  and  $\bar{\eta}_i$  (divided by  $|\varepsilon|$ ), which describe the power-law decay of correlations in the QLRO phase of the RF model below the lower critical dimension, as functions of  $N$  for  $N < N_c$ . **Right panel:** the critical exponents  $\eta_i$  and  $\bar{\eta}_i$  (divided by  $\varepsilon$ ), which describe the paramagnetic-ferromagnetic transition of the RF model above the lower critical dimension, as functions of  $N$  for  $N > N_c$ . **Inset:** the bulk magnetization exponent  $\beta$  and the surface magnetization exponent  $\beta_1$  as functions of  $N$ .

are shown in the inset of the right panel of Figure 1. To one-loop order they obey the relation  $\beta_1 = 2\beta$ . Up to now both magnetization exponents have been studied only for the 3D RFIM where numerical simulations give  $\beta = 0.0017 \pm 0.005$  [25] and  $\beta_1 = 0.23 \pm 0.03$  [21]. Thus, the ratio  $\beta_1/\beta$  for the RF  $O(N)$  systems in  $d > 4$  is much smaller than for the 3D RFIM.

#### 4.2. Quasi-long-range order for $d < 4$ ( $\varepsilon < 0$ )

Below the lower critical dimension the flow equation for the disorder correlator has an attractive  $2\pi$ -periodic FP solution. This cuspy FP appears only for  $2 \leq N < N_c$  where it controls the scaling behavior of spin fluctuations in the QLRO phase. The corresponding exponents  $\eta_i$  and  $\bar{\eta}_i$  as functions of  $N$  are shown in the left panel of Figure 1. In the case  $N = 2$  the FP equation admits for an explicit non-analytic  $\phi_0$ -periodic solution given by

$$R^*(\phi) = \frac{|\varepsilon|\phi_0^4}{72} \left[ \frac{1}{36} - \left( \frac{\phi}{\phi_0} \right)^2 \left( 1 - \frac{\phi}{\phi_0} \right)^2 \right]. \quad (49)$$

Using equations (46)-(48) one obtains

$$\eta = \frac{\phi_0^2}{36} |\varepsilon|, \quad \bar{\eta} = \left( 1 + \frac{\phi_0^2}{36} \right) |\varepsilon|, \quad \bar{\eta}_{\perp} = \left( 1 + \frac{\phi_0^2}{24} \right) |\varepsilon|, \quad (50)$$

$$\eta_{\parallel} = \frac{\phi_0^2}{18} |\varepsilon|, \quad \bar{\eta}_{\parallel} = \left( 1 + \frac{\phi_0^2}{18} \right) |\varepsilon|, \quad \eta_{\perp} = \frac{\phi_0^2}{24} |\varepsilon| \quad (51)$$

with  $\phi_0 = 2\pi$  for the RF system. The semi-infinite RF  $O(2)$  model can be mapped onto a semi-infinite periodic disordered elastic system with a free surface. There is one to one correspondence between the Bragg glass phase of the elastic system

and the QLRO phase of the studied spin model. The power-law decay of the spin correlations in the QLRO phase corresponds to the logarithmic growth of the displacements in the disordered elastic system. Moreover, the exponents  $\eta$ ,  $\eta_{\perp}$  and  $\eta_{\parallel}$  provide the universal amplitudes of the logarithmic growth of the displacements in the bulk, at the surface and along the surface, respectively. For a  $\phi_0$ -periodic elastic system with a free surface these amplitudes are given by equations (50)-(51). In particular, we find that the logarithmic growth of the displacements along the surface is twice large as the logarithmic growth in the bulk. In the case when only one point is on the surface the growth is enhanced by 50%. The presence of a free surface can be considered as an extended defect of a special kind. The influence of potential-like extended defects on the Bragg-glass has been recently studied in [10,27].

## 5. Summary

In the present work we have studied the RF semi-infinite  $O(N)$  systems with a free surface. Above the lower critical dimension  $d_{lc} = 4$  the systems undergo a paramagnetic-ferromagnetic transition for  $N > N_c$ , while below  $d_{lc}$  and for  $N < N_c = 2.835$  they exhibit a QLRO phase with zero magnetization and power-law correlation of fluctuations. Using FRG we have derived the surface scaling behavior at criticality as well as in the QLRO phase, and calculated the corresponding surface exponents to lowest order in  $\varepsilon = d - 4$ . We have found that the dimensional reduction prediction for the surface scaling is broken similar to what happens in the bulk. We have shown that the connected and disconnected correlation functions scale differently also at the surface and derived the scaling relations between different surface exponents. The surface exponents obtained for the 3D RF  $O(2)$  can be used to describe the growth of displacements near a free surface in semi-infinite periodic elastic systems in disordered media. The methods developed in this work can be also applied to the systems with random anisotropy disorder [8].

## REFERENCES

1. Binder K. Phase Transitions and critical phenomena / Edited by C. Domb and J. L. Lebowitz. London : Academic, 1983.
2. Blatter G., Feigel'man M.V., Geshkenbein V.B., Larkin A.I., Vinokur V.M. Vortices in high-temperature superconductors // Rev. Mod. Phys. 1994. V. 66, Issue 4. P. 1125–1388.
3. Clark N.A., Bellini T., Malzbender R.M., Thomas B.N., Rappaport A.G., Muzny C.D., Schaefer D.W., Hrubesh L. X-ray scattering study of smectic ordering in a silica aerogel // Phys. Rev. Lett. 1993. V. 71, Issue 21. P. 3505–3508.
4. Diehl H.W. Phase transitions and critical phenomena / Edited by C. Domb and J.L. Lebowitz. London : Academic, 1986.
5. Diehl H.W., Nüsser A. Critical behavior of the nonlinear  $\sigma$  model with a free surface: The “ordinary” transition in  $2+\epsilon$  dimensions // Phys. Rev. Lett. 1986. V. 56, Issue 26. P. 2834–2837.

6. Dudka M., Folk R., Holovatch Yu. Critical properties of random anisotropy magnets // *Journal of Magnetism and Magnetic Materials*. 2005. V. 294., N. 3. P. 305–329.
7. Elbs J., Bunkov Yu.M., Collin E., Godfrin H., Volovik G.E. Strong orientational effect of stretched aerogel on the  $^3\text{He}$  order parameter // *Phys. Rev. Lett.* 2008. V. 100, Issue 21. P. 215304.
8. Fedorenko A.A. Random-field and random-anisotropy  $O(N)$  spin systems with a free surface // *Phys. Rev. E*. 2012. V. 86, Issue 2. P. 021131.
9. Fedorenko A.A., Kühnel F. Long-range correlated random field and random anisotropy  $O(N)$  models: A functional renormalization group study // *Phys. Rev. B*. 2007. V. 75, Issue 17. P. 174206.
10. Fedorenko A.A., Le Doussal P., Wiese K.J. Statics and dynamics of elastic manifolds in media with long-range correlated disorder // *Phys. Rev. E*. 2006. V. 74, Issue 6. P. 061109.
11. Fedorenko A.A., Trimper S. Critical aging of a ferromagnetic system from a completely ordered state // *Europhys. Lett.* 2006. V. 74. P. 89–95.
12. Feldman D.E. Quasi-long-range order in nematics confined in random porous media // *Phys. Rev. Lett.* 2000. V. 84, Issue 21. P. 4886–4889.
13. Feldman D.E. Quasi-long-range order in the random anisotropy Heisenberg model: Functional renormalization group in  $4 - \epsilon$  dimensions // *Phys. Rev. B*. 2000. V. 61, Issue 1. P. 382–390.
14. Feldman D.E., Vinokur V.M. Destruction of bulk ordering by surface randomness // *Phys. Rev. Lett.* 2002. V. 89, Issue 22. P. 227204.
15. Fisher D.S. Random fields, random anisotropies, nonlinear  $\sigma$  models, and dimensional reduction // *Phys. Rev. B*. 1985. V. 31, Issue 11. P. 7233–7251.
16. Fisher D.S. Scaling and critical slowing down in random-field Ising systems // *Phys. Rev. Lett.* 1986. V. 56, Issue 5. P. 416–419.
17. Fridrikh S.V., Terentjev E.M. Order-disorder transition in an external field in random ferromagnets and nematic elastomers // *Phys. Rev. Lett.* 1997. V. 79, Issue 23. P.4661–4664.
18. Harris R., Plischke M., Zuckermann M.J. New model for amorphous magnetism // *Phys. Rev. Lett.* 1973. V. 31, Issue 3. P. 160–162.
19. Imry Y., Ma S. Random-field instability of the ordered state of continuous symmetry // *Phys. Rev. Lett.* 1975. V. 35, Issue 21. P. 1399–1401.
20. Kühnel F., Schwarz D.J. Stochastic inflation and dimensional reduction // *Phys. Rev. D*. 2008. V. 78, Issue 10. P. 103501.
21. Laurson L., Alava M.J. Surface criticality in random field magnets // *Phys. Rev. B*. 2005. V. 72, Issue 21. P. 214416.
22. Le Doussal P., Wiese K.J. Random-field spin models beyond 1 loop: A mechanism for decreasing the lower critical dimension // *Phys. Rev. Lett.* 2006. V. 96, Issue 19. P. 197202.
23. Lubensky T.C., Rubin M.H. Critical phenomena in semi-infinite systems. II. Mean-field theory // *Phys. Rev. B*. 1975. V. 12, Issue 9. P. 3885–3901.
24. Matsumoto K., Porto J.V., Pollack L., Smith E.N., Ho T.L., Parpia J.M. Quantum phase transition of  $^3\text{He}$  in aerogel at a nonzero pressure // *Phys. Rev. Lett.* 1997. V. 79, Issue 2. P. 253–256.
25. Middleton A.A., Fisher D.S. Three-dimensional random-field Ising magnet: Interfaces,

- 
- scaling, and the nature of states // *Phys. Rev. B.* 2002. V. 65, Issue 13. P. 134411.
26. Nattermann T. Spin glasses and random fields / Edited by A.P. Young. Singapore : World Scientific, 1998.
  27. Petković A., Emig T., Nattermann T. Pinning of flux lines by planar defects // *Phys. Rev. B.* 2009. V. 79, Issue 22. P. 224512.
  28. Pleimling M. Critical phenomena at perfect and non-perfect surfaces // *J. Phys. A: Math. Gen.* 2004. V. 37. P. R79–R115.
  29. Saber M. Mean-field theory of the semi-infinite spin-4 Ising model with random surface and bulk fields // *J. Phys. C.* 1987. V. 20. P. 2749.
  30. Sakamoto Y., Mukaida H., Itoi C. Effect of second-rank random anisotropy on critical phenomena of a random-field  $O(N)$  spin model in the large  $N$  limit // *Phys. Rev. B.* 2005. V. 72, Issue 14. P. 144405.
  31. Tarjus G., Tissier M. Nonperturbative functional renormalization group for random-field models: The way out of dimensional reduction // *Phys. Rev. Lett.* 2004. V. 93, Issue 26. P. 267008.
  32. Tissier M., Tarjus G. Supersymmetry and its spontaneous breaking in the random field Ising model // *Phys. Rev. Lett.* 2011. V. 107, Issue 4. P. 041601.
  33. Usatenko Z., Spalek J. Surface critical behavior of semi-infinite systems with cubic anisotropy at the ordinary transition // *J. Phys. A: Math. Gen.* 2004. V. 37. P. 7113–7125.

## **BRANS-DICKE SCALAR-TENSOR THEORY OF GRAVITATION MAY EXPLAIN TIME ASYMMETRY OF PHYSICAL PROCESSES**

**O. Kosheleva, V. Kreinovich**

Most fundamental physical equations remain valid if we reversed the time order. Thus, if we start with a physical process (which satisfies these equations) and reverse time order, the resulting process also satisfies all the equations and thus, should also be physically reasonable. In practice, however, many physical processes are not reversible: e.g., a cup can break into pieces, but the pieces cannot magically get together and become a whole cup. In this paper, we show that the Brans-Dicke Scalar-Tensor Theory of Gravitation, one of the most widely used generalizations of Einstein's General relativity, is, in effect, time-asymmetric. This time-asymmetry may explain the observed time asymmetry of physical phenomena.

### **1. Time Asymmetry: Formulation of the Problem**

**Observable time asymmetry: a problem.** Most equations of fundamental physics are time symmetric, starting from the ordinary differential equations (e.g., the classical Newton's equations of motion) to partial differential equations describing physical fields like electromagnetism or gravitation. As a result, if we start with a physically reasonable solution to these equations (e.g., with the observed Universe) and simply reverse the direction of time  $t$ , the resulting fields will satisfy the same differential equations. From this theoretical viewpoint, all physical processes should be reversible: a time reversal of a physically reasonable process should also be physically reasonable.

In practice, however, many physical processes are not reversible. For example:

- If we drop a fragile cup, it will break into pieces.
- However, it is not physically reasonable to expect that the pieces of a broken cup would magically get together to form a whole cup.

**How this problem is explained now.** The problem of time asymmetry is known since Boltzmann's 19th century work on statistical physics and its foundations. In modern physics, this problem is usually resolved by making an additional assumption: that the initial conditions should be *random* (in some reasonable sense); see, e.g., [1].

**Limitations of the known explanation.** The additional assumption of randomness is outside the usual formulation of physical equations as a system of partial differential equations. It is therefore desirable to come up with an alternative explanation of the observed time asymmetry, an explanation that is within the usual formulation.

**What we do in this paper.** In this paper, we show that the time asymmetry problem can be potentially resolved if we take into account that for scalar-tensor theories of gravitation, in some reasonable sense, equations are *not* T-symmetric.

## 2. Brans-Dicke Theory Scalar-Tensor Theory of Gravitation: Reminder

*Notational comment.* Before we describe the actual equations, let us agree to simplifying notations which are commonly used in General Relativity.

The possibility of this simplification is based on the fact that, according to Relativity theory, the speed of light  $c$  is a universal constant. Thus:

- If we fix a unit of time (e.g., 1 sec), we automatically get a unit of length — namely, the distance that light can cover in 1 sec.
- Similarly, once we have a unit of length, we get a unit of time.

Relativistic equations can be simplified if we use units of distance and time which are related in this way, i.e., units in which the speed of light is simply equal to 1. For simplicity, we will use these units in our paper.

**General Relativity: reminder.** In general, the field equations of a physical theory correspond to the minimum of the action

$$S = \int L\sqrt{-g} dt dV,$$

where  $L$  is the Lagrangian and  $g = \det(g_{\alpha\beta})$  is the determinant of the metric tensor  $g_{\alpha\beta}$ . In other words, a physical theory corresponds to the variation principle

$$\delta \int L\sqrt{-g} dt dV = 0.$$

In particular, for the General Relativity theory (GRT), the Lagrangian has the form

$$L_{GRT} = \frac{1}{16\pi G} R + L_{\text{mat}},$$

where  $G$  is the gravitation constant,  $L_{\text{mat}}$  is the Lagrangian of matter,  $R \stackrel{\text{def}}{=} g^{\alpha\beta} R_{\alpha\beta}$  is the Ricci scalar,  $R_{\alpha\beta} \stackrel{\text{def}}{=} R_{\alpha\gamma\beta}^{\gamma}$ , and  $R_{\alpha\gamma\beta}^{\delta}$  is the curvature tensor. The variational equations do not change if we simply multiply the Lagrangian by a constant; it is therefore useful to multiply the Lagrangian by  $16\pi$  and take

$$L_{GRT} = \frac{1}{G} R + 16\pi L_{\text{mat}}.$$

Varying over  $g_{\alpha\beta}$ , we get the corresponding equations

$$R_{\alpha\beta} - \frac{1}{2} g_{\alpha\beta} R = 8\pi G T_{\alpha\beta},$$

where  $T_{\alpha\beta}$  is the matter's energy-momentum tensor.

**Motivations for modifying General Relativity.** According to General Relativity, space-time and gravitation are described by Einstein's partial differential equations, in which the only field responsible for gravitation is the metric tensor  $g_{\alpha\beta}$ . In General Relativity — similarly to the original Newton's theory of gravitation — the gravitational field (and hence, gravitational acceleration  $a$ ) generated by a body of mass  $M$  at a distance  $r$  is proportional to its mass  $M$ , with the gravitation constant  $G$  as the proportionality coefficient:  $a \approx \frac{GM}{r^2}$ .

It turns out that the observed gravitational accelerations in the vicinity of several distant astronomical bodies are much larger than what is predicted based on the observable mass  $M_{\text{obs}}$ :  $a \gg \frac{GM}{r^2}$ . The traditional approach to this problem is to conclude that, in addition to the observable masses, there are also non-observable ones. In this approach, to explain the observations, we must assume that on the cosmological level, 95% of the mass is formed by hypothetical non-directly-observable “dark matter” and “dark energy”.

Some physicists argue that instead of introducing such hypothetical types of matter, it is more reasonable to conclude that the parameter  $G$  that described the local strength of gravitational interactions does not have to be a universal constant: measurements of  $G$  at different points in space-time can lead, in general, to different results. In effect, the values  $\varphi(x)$  of  $\frac{1}{G}$  measured at different space-time points  $x$  form a new scalar field. So, in such a theory, to describe the gravitational field, we need to present both the metric field  $g_{\alpha\beta}$  and the scalar field  $\varphi$ .

The corresponding scalar-tensor theory of gravitation was indeed proposed by Brans and Dicke (see, e.g., [2], Chapter 39); the equations of this theory are presented below.

*Historical comment.* Historically the first modification of General Relativity, in which there is no need for the hypothetical dark energy and dark matter, came in the form of a modified Lagrangian which only depends on the metric  $g_{\alpha\beta}$  but which — in contrast to the Lagrangian of General Relativity — is non-linear in the scalar curvature  $R$ . However, a recent paper [3] showed that such theories are equivalent to scalar-tensor theories of Brans-Dicke type.

**Brans-Dicke Theory: reminder.** In the scalar-tensor theory of gravitation, the parameter  $G$  that described the local strength of gravitational interactions is no longer a universal constant, it is equal to  $1/\varphi$ , where  $\varphi$  is the new scalar field. In terms of this new field, the Einstein's term  $\frac{1}{G}R$  from the Lagrangian takes the form  $\varphi R$ .

To get a full description of the scalar-tensor theory, we also need to add, to the Lagrangian, the term  $\frac{\varphi_{,\alpha}\varphi^{,\alpha}}{\varphi}$  describing the effective energy density of the scalar field. As a result, we arrive at the following Lagrangian:

$$L_{\text{BDT}} = \varphi \left( R - \frac{\varphi_{,\alpha}\varphi^{,\alpha}}{\varphi^2} \right) + 16\pi L_{\text{mat}}.$$

Varying over  $g_{\alpha\beta}$  and  $\varphi$ , we get the following equations:

$$R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R = \frac{8\pi}{\varphi}T_{\alpha\beta} + \frac{\omega}{\varphi^2} \left( \varphi_{,\alpha}\varphi_{,\beta} - \frac{1}{2}g_{\alpha\beta}\varphi_{,\gamma}\varphi^{,\gamma} \right) + \frac{1}{\varphi}(\varphi_{;\alpha\beta} - g_{\alpha\beta}\square\varphi); \quad (1)$$

$$\square\varphi = \varphi_{;\alpha}^{\alpha} = \frac{8\pi}{3+2\omega}T, \quad (2)$$

where  $T \stackrel{\text{def}}{=} T_{\alpha}^{\alpha}$  is the trace of the energy-momentum tensor [2].

**At first glance, the Brans-Dicke theory is T-symmetric.** At first glance, from the viewpoint of time symmetry, the Brans-Dicke Theory (BDT) is similar to Einstein's General Relativity:

- similar to General Relativity, the Brans-Dicke Theory is described by second order partial differential equations, and
- the BDT equations remain invariant if we reserve the order of time  $t$ , i.e., change  $t$  to  $-t$ .

In general, in a second-order theory,

- if on some Cauchy surface (e.g., for some moment of time  $t_0$ ), we know the values of the gravity tensor  $g_{\alpha\beta}$ , the scalar field  $\varphi$ , and their first time derivatives  $\dot{g}_{\alpha\beta}$  and  $\dot{\varphi}$ ,
- then we can uniquely determine the second time derivatives  $\ddot{g}_{\alpha\beta}$  and  $\ddot{\varphi}$ , and thus (at least locally) integrate the corresponding equations.

### 3. Main Result: Cauchy Problem for the Brans-Dicke Scalar-Tensor Theory of Gravitation Leads to T-Asymmetry

**Our main result: formulation.** In this section, we prove a new (somewhat unexpected) result: that with respect to the scalar field  $\varphi$ , the Brans-Dicke scalar theory of gravitation is actually *first order*. Specifically:

- if on some Cauchy surface, we know the values of the gravity tensor  $g_{\alpha\beta}$ , its first time derivative  $\dot{g}_{\alpha\beta}$ , and the field  $\varphi$ ,
- then we can determine the first time derivative  $\dot{\varphi}$  of  $\varphi$  from a quadratic equation.

This quadratic equation, in general, has two solutions. This means that in principle, for each initial condition, we can have *two* different dynamics — corresponding to these two solutions.

**Discussion.** In physical terms, our result means that the time-symmetric Brans-Dicke Theory (BDT), in effect, consists of *two* different theories — each of which is second order in metric tensor and first order in  $\varphi$ .

Each solution of BDT is a solution of one of these two theories. In particular, our Universe satisfies one of the corresponding two systems of partial differential equations.

The transformation  $t \rightarrow -t$  transforms each of these two theories into another one, but none of these two theories is time-symmetric. In other words, in the presence of the additional scalar gravitational field, the equations describing our Universe are *not* time symmetric.

This may explain the observed time asymmetry of physical phenomena.

*Historical comment.* The fact that the first derivative  $\dot{\varphi}$  is not needed was first shown in Section 16.4 of [4] for homogeneous isotropic cosmological solutions. In this paper, we show that it is possible to describe  $\dot{\varphi}$  in terms of other initial conditions  $\varphi$ ,  $g_{\alpha\beta}$ , and  $\dot{g}_{\alpha\beta}$  in the general case as well.

**How we prove our result.** In the following text, we will transform the Brans-Dicke equations step-by-step. After these transformation, we will see that the first time derivative  $\dot{\varphi}$  of the scalar field  $\varphi$  on a Cauchy surface can indeed be determined in terms of the values of the metric field  $g_{\alpha\beta}$ , its first time derivative  $\dot{g}_{\alpha\beta}$ , and the field  $\varphi$  on this surface.

**First transformation: into equivalent equations for the Ricci tensor.** The equations (1) describe the Einstein tensor  $R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R$ . Let us first use these

equations to describe the Ricci tensor  $R_{\alpha\beta}$ . By definition,

$$R_{\alpha\beta} = \left( R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R \right) + \frac{1}{2}g_{\alpha\beta}R; \tag{3}$$

so, to describe the Ricci tensor, it is sufficient to be able to describe the Ricci scalar  $R$ . If we take the diagonal of the equation (1), we get

$$-R = \frac{8\pi}{\varphi}T - \frac{\omega}{\varphi^2}\varphi_{,\alpha}\varphi^{,\alpha} - \frac{3}{\varphi}\square\varphi,$$

so

$$R = -\frac{8\pi}{\varphi}T + \frac{\omega}{\varphi^2}\varphi_{,\alpha}\varphi^{,\alpha} + \frac{3}{\varphi}\square\varphi. \tag{4}$$

Substituting this expression (4) for  $R$  and the expression (1) for  $R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R$  into the formula (3), we get

$$\begin{aligned} R_{\alpha\beta} &= \frac{8\pi}{\varphi}T_{\alpha\beta} - \frac{4\pi}{\varphi}Tg_{\alpha\beta} + \frac{\omega}{\varphi^2} \left( \varphi_{,\alpha}\varphi_{,\beta} - \frac{1}{2}g_{\alpha\beta}\varphi_{,\gamma}\varphi^{,\gamma} \right) + \frac{1}{2}\frac{\omega}{\varphi^2}\varphi_{,\alpha}\varphi^{,\alpha} + \\ &\quad + \frac{1}{\varphi}(\varphi_{;\alpha\beta} - g_{\alpha\beta}\square\varphi) + \frac{3}{2\varphi}\square\varphi = \\ &= \frac{8\pi}{\varphi} \left( T_{\alpha\beta} - \frac{1}{2}Tg_{\alpha\beta} \right) + \omega\frac{\varphi_{,\alpha}\varphi_{,\beta}}{\varphi^2} + \frac{\varphi_{;\alpha\beta}}{\varphi} + \frac{1}{2}\frac{\square\varphi}{\varphi}g_{\alpha\beta}. \end{aligned}$$

Substituting the expression (2) for  $\square\varphi$  into this formula, we conclude that

$$R_{\alpha\beta} = \frac{8\pi}{\varphi} \left( T_{\alpha\beta} - \frac{1+\omega}{3+2\omega}Tg_{\alpha\beta} \right) + \omega\frac{\varphi_{,\alpha}\varphi_{,\beta}}{\varphi^2} + \frac{\varphi_{;\alpha\beta}}{\varphi}. \tag{5}$$

**Second transformation: into Gaussian normal coordinates.** We will use Gaussian normal coordinates, in which  $g_{00} = 1$  and  $g_{0i} = 0$  for  $i = 1, 2, 3$ , so only the values  $g_{ij}$  corresponding to  $i, j = 1, 2, 3$  are changing. In these coordinates, the distance element  $ds^2 = g_{\alpha\beta}dx^\alpha dx^\beta$  takes the form

$$ds^2 = dt^2 - \gamma_{ij}dx^i dx^j,$$

where we denoted  $\gamma_{ij} \stackrel{\text{def}}{=} -g_{ij}$ . In the Gaussian normal coordinates, the components  $R_{00}$ ,  $R_{0i}$ , and  $R_{ij}$  of the Ricci tensor  $R_{\alpha\beta}$  can be explicitly described in terms of  $\gamma_{ij}$ ,  $\varkappa_{ij} \stackrel{\text{def}}{=} \dot{g}_{ij} = -\dot{\gamma}_{ij}$ , and the time derivative  $\dot{\varkappa}_{ij}$ ; see, e.g., [2]. Substituting these expressions for  $R_{00}$ ,  $R_{0i}$ , and  $R_{ij}$  into the formula (5), and using similar expressions for the derivatives of  $\varphi$ , we get the following equations:

$$-\frac{1}{2}\dot{\varkappa}_i^i - \frac{1}{4}\varkappa_j^i \varkappa_i^j = \frac{8\pi}{\varphi} \left( T_{00} - \frac{1+\omega}{3+2\omega}T \right) + \omega\frac{(\dot{\varphi})^2}{\varphi^2} + \frac{\ddot{\varphi}}{\varphi}; \tag{6}$$

$$\frac{1}{2}\varkappa_{i;j}^j - \frac{1}{2}\varkappa_{j;i}^j = \frac{8\pi}{\varphi}T_{0i} + \omega\frac{\dot{\varphi}\varphi_{,i}}{\varphi^2} + \frac{\dot{\varphi}_{,i}}{\varphi}, \tag{7}$$

$$\begin{aligned}
 & P_{ij} - \frac{1}{2}\dot{\varkappa}_{ij} - \frac{1}{4}(\varkappa_{ij}\varkappa_k^k - 2\varkappa_i^k\varkappa_{kj}) = \\
 & = \frac{8\pi}{\varphi} \left( T_{ij} + \frac{1+\omega}{3+2\omega} T\gamma_{ij} \right) + \omega \frac{\varphi_{,i}\varphi_{,j}}{\varphi^2} + \frac{\varphi_{;ij} - \varkappa_{ij}\dot{\varphi}}{\varphi}, \tag{8}
 \end{aligned}$$

where  $P_{ij}$  is the 3-D curvature tensor, and all tensor operations are performed in the space with metric  $\gamma_{ij}$ .

The equation (2) takes the form

$$\ddot{\varphi} - \Delta\varphi = \frac{8\pi}{3+2\omega} T, \tag{9}$$

where  $\Delta\varphi \stackrel{\text{def}}{=} \varphi_{;i}^i$  is the 3-D Laplace operator.

**It is possible to describe  $\dot{\varphi}$  in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ : idea of the proof.**

The above equations describe the second time derivatives of  $\varphi$  and  $\gamma_{ij}$  in terms of the values  $\varphi$  and  $\gamma_{ij}$  and their first time derivatives.

Let us show that we can actually describe the *first* derivative  $\dot{\varphi}$  in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ . Indeed:

- from the equation (9), we can explicitly express  $\ddot{\varphi}$  in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ ;
- from the equation (8), we can explicitly express  $\dot{\varkappa}_{ij}$  (and, thus,  $\dot{\varkappa}_i^i$ ) in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ ,  $\varphi$ , and  $\dot{\varphi}$ ; the resulting dependence of  $\dot{\varkappa}_i^i$  on  $\dot{\varphi}$  is linear.

Substituting these expression for  $\dot{\varkappa}_i^i$  and  $\ddot{\varphi}$  into the equality (6), we get a quadratic equation for  $\dot{\varphi}$ . This quadratic equation allows us to determine  $\dot{\varphi}$  in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ .

Since the equation is quadratic, for each combination of initial values  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ , we may get *two* possible values of  $\dot{\varphi}$ .

**How to describe  $\dot{\varphi}$  in terms of  $\gamma_{ij}$ ,  $\dot{\gamma}_{ij}$ , and  $\varphi$ : details.** From the equation (9), we can conclude that

$$\ddot{\varphi} = \Delta\varphi + \frac{8\pi}{3+2\omega} T. \tag{10}$$

Similarly, from the equation (8), we conclude that

$$\begin{aligned}
 -\frac{1}{2}\dot{\varkappa}_{ij} & = -P_{ij} + \frac{1}{4}(\varkappa_{ij}\varkappa_k^k - 2\varkappa_i^k\varkappa_{kj}) + \frac{8\pi}{\varphi} \left( T_{ij} + \frac{1+\omega}{3+2\omega} T\gamma_{ij} \right) + \\
 & + \omega \frac{\varphi_{,i}\varphi_{,j}}{\varphi^2} + \frac{\varphi_{;ij} - \varkappa_{ij}\dot{\varphi}}{\varphi}, \tag{11}
 \end{aligned}$$

Hence, we get

$$-\frac{1}{2}\dot{\varkappa}_i^i = -P + \frac{1}{4} \left( (\varkappa_i^i)^2 - 2\varkappa_j^i\varkappa_i^j \right) + \frac{8\pi}{\varphi} \frac{6+5\omega}{3+2\omega} T + \omega \frac{\varphi_{,i}\varphi^{,i}}{\varphi^2} + \frac{\Delta\varphi - \varkappa_i^i\dot{\varphi}}{\varphi}. \tag{12}$$

Substituting formulas (10) and (12) into the equation (6) and moving all the terms to the right-hand side, we get the desired quadratic equation

$$A(\dot{\varphi})^2 + B\dot{\varphi} + C = 0, \tag{13}$$

where

$$\begin{aligned} A &\stackrel{\text{def}}{=} \frac{\omega}{\varphi^2}; \quad B \stackrel{\text{def}}{=} \frac{\varkappa_i^i}{\varphi}; \quad \text{and} \\ C &\stackrel{\text{def}}{=} \frac{1}{4} \varkappa_j^i \varkappa_i^j + \frac{8\pi}{\varphi} \left( T_{00} - \frac{1+\omega}{3+2\omega} T \right) + \frac{\Delta\varphi}{\varphi} + \frac{8\pi}{\varphi} \frac{1}{3+2\omega} T + P - \\ &\quad - \frac{1}{4} \left( (\varkappa_i^i)^2 - 2\varkappa_j^i \varkappa_i^j \right) - \frac{8\pi}{\varphi} \frac{6+5\omega}{3+2\omega} T - \omega \frac{\varphi_{,i} \varphi^{,i}}{\varphi^2} - \frac{\Delta\varphi}{\varphi} = \\ &\quad - \frac{8\pi}{\varphi} \left( T_{00} - \frac{6+6\omega}{3+2\omega} T \right) + P + \frac{3}{4} (\varkappa_i^i)^2 - \frac{1}{4} \varkappa_j^i \varkappa_i^j - \omega \frac{\varphi_{,i} \varphi^{,i}}{\varphi^2}. \end{aligned} \tag{15}$$

*Mathematical comment.* When the quadratic equation has two solutions  $\dot{\varphi}_1$  and  $\dot{\varphi}_2$  for  $\dot{\varphi}$ , we can substitute both solutions into the equation (7) and take the difference between the resulting equalities. Then, for the difference  $D \stackrel{\text{def}}{=} \dot{\varphi}_1 - \dot{\varphi}_2$ , we get the equality

$$\omega \frac{D\varphi_{,i}}{\varphi^2} + \frac{D_{,i}}{\varphi} = 0.$$

Multiplying both sides of this equation by  $\varphi$  and dividing by  $d$ , we conclude that

$$\omega \frac{\varphi_{,i}}{\varphi} + \frac{D_{,i}}{D} = 0,$$

i.e., equivalently,

$$\omega(\ln(\varphi))_{,i} + (\ln(D))_{,i} = (\omega \ln(\varphi) + \ln(D))_{,i} = 0.$$

Since the gradient of the expression  $\omega \ln(\varphi) + \ln(D)$  is equal to 0, this expression is constant, so  $\omega \ln(\varphi) + \ln(D) = c$ ,  $\ln(D) = c - \omega \ln(\varphi)$  and thus,

$$\dot{\varphi}_1 - \dot{\varphi}_2 = D = c_1 \cdot \varphi^{-\omega},$$

where we denoted  $c_1 \stackrel{\text{def}}{=} e^c$ .

**The same T-asymmetry holds for more general scalar-tensor theories of gravitation.** The authors of [3] also consider generalizations of Brans-Dicke theory, with the Lagrangian

$$L = \varphi \left( R - \frac{\varphi_{,\alpha} \varphi^{,\alpha}}{\varphi^2} - V(\varphi) \right) + 16\pi L_{\text{mat}},$$

where  $V(\varphi)$  is the potential of the scalar field. The original Brans-Dicke theory is a particular case of this general theory, corresponding to  $V(\varphi) = 0$ . For this

Lagrangian, the variational equations (generalizing equations (1) and (2)) take the form

$$R_{\alpha\beta} - \frac{1}{2}g_{\alpha\beta}R = \frac{8\pi}{\varphi}T_{\alpha\beta} + \frac{\omega}{\varphi^2} \left( \varphi_{,\alpha}\varphi_{,\beta} - \frac{1}{2}g_{\alpha\beta}\varphi_{,\gamma}\varphi^{,\gamma} \right) + \frac{1}{\varphi}(\varphi_{;\alpha\beta} - g_{\alpha\beta}\square\varphi) - \frac{1}{2}\frac{V(\varphi)}{\varphi}g_{\alpha\beta}; \quad (16)$$

$$\square\varphi = \varphi^{;\alpha}_{;\alpha} = \frac{8\pi}{3+2\omega}T - \frac{1}{3+2\omega} \left( V - \varphi\frac{dV}{d\varphi} \right). \quad (17)$$

The two additional terms depend only on  $\varphi$ , so, as one can see, they do not change the fact that the first derivative  $\dot{\varphi}$  can be (almost) uniquely determined by the initial values of  $\varphi$ ,  $g_{ij}$ , and  $\dot{g}_{ij}$ .

*Comment.* It is worth mentioning that a similar possibility to reconstruct  $\dot{\varphi}$  from the initial values of  $\varphi$ ,  $g_{ij}$ , and  $\dot{g}_{ij}$  is also available for more traditional scalar-tensor theories, with a Lagrangian of the type

$$L = \frac{1}{G}R + L_{\text{scalar}}(\varphi, \varphi_{\alpha}\varphi^{,\alpha}) + 16\pi L_{\text{mat}}.$$

For these theories, the proof of this reconstruction possibility is even easier than for the Brans-Dicke-type theories. Indeed, for such more traditional scalar-tensor theories, the right-hand side of the corresponding Einstein equations  $R_{\alpha\beta} - \frac{1}{2}Rg_{\alpha\beta} = \dots$  depends only on the first derivatives  $\dot{\varphi}$  and  $\varphi_{,i}$  of the scalar field. In particular, the right-hand side of the equation corresponding to  $R_{0i}$  (similar to our equation (7)) contains only  $\dot{\varphi}$  and thus, can be used to explicitly express  $\dot{\varphi}$  in terms of  $\varphi$ ,  $g_{ij}$  and  $\dot{g}_{ij}$ .

## 4. Conclusion

Our result shows that the time-symmetric Brans-Dicke Theory of gravitation (BDT), in effect, consists of two different theories — each of which is second order in metric tensor and first order in  $\varphi$ . Each solution of BDT is a solution of one of these two theories. In particular, our Universe satisfies one of the corresponding two systems of partial differential equations. The transformation  $t \rightarrow -t$  transforms each of these two theories into another one, but none of these two theories is time-symmetric.

In other words, in the presence of the additional scalar gravitational field, the equations describing our Universe are *not* time symmetric. This may explain the observed time asymmetry of physical phenomena.

## Acknowledgments

This research was partly supported by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-

0926721, by Grant 1 T36 GM078000-01 from the National Institutes of Health, and by a grant on F-transforms from the Office of Naval Research.

## REFERENCES

1. Feynman R., Leighton R., Sands M. The Feynman Lectures on Physics. Boston, Massachusetts : Addison Wesley, 2005.
2. Misner C.W., Thorne K.S., Wheeler J.A. Gravitation. New York : W.H. Freeman, 1973.
3. Olmo G.J., Sanchis-Alepus H. Hamiltonian formulation of Palatini  $f(R)$  theories á la Brans-Dicke theory // Phys. Rev. D. 2011. V. 83, N. 10, Publ. 104036.
4. Weinberg S. Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity. New York: J. Wiley, 1972.

## **FOR DESCRIBING UNCERTAINTY, ELLIPSOIDS ARE BETTER THAN GENERIC POLYHEDRA AND PROBABLY BETTER THAN BOXES: A REMARK**

**O. Kosheleva, V. Kreinovich**

For a single quantity, the set of all possible values is usually an interval. An interval is easy to represent in a computer: e.g., we can store its two endpoints. For several quantities, the set of possible values may have an arbitrary shape. An exact description of this shape requires infinitely many parameters, so in a computer, we have to use a finite-parametric approximation family of sets. One of the widely used methods for selecting such a family is to pick a symmetric convex set and to use its images under all linear transformations. If we pick a unit ball, we end up with ellipsoids; if we pick a unit cube, we end up with boxes and parallelepipeds; we can also pick a polyhedron. In this paper, we show that ellipsoids lead to better approximations of actual sets than generic polyhedra; we also show that, under a reasonable conjecture, ellipsoids are better approximators than boxes.

### **1. Formulation of the Problem**

**Need for describing sets of possible values.** Measurement and estimates are never 100% accurate. As a result, we usually do not know the exact value of a physical quantity; we usually know the set of possible values of this quantity. For a single quantity, this set is usually an interval. Representing an interval in a computer is easy: e.g., we can represent an interval by its endpoints; see, e.g., [7, 10].

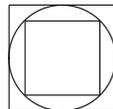
For several quantities  $x_1, \dots, x_n$ , in addition to interval bounds on each of these quantities, we often have additional restrictions on their combinations; as a result, the set of possible values of  $x = (x_1, \dots, x_n)$  can have different shapes. The space of all possible sets is infinite-dimensional, meaning that we need infinitely many real-valued parameters to represent a generic set. In a computer, at any given time, we can only store finitely many parameters; so, we cannot represent generic sets exactly, we need to approximate them by sets from a finite-parametric family.

**Convex set-based representation of sets of possible values.** In many practical situations, e.g., when  $x_i$  are spatial coordinates, the selection of the quantities is rather arbitrary: we can use a different coordinate system in which, instead of the original quantities  $x_i$ , we use linear combinations  $y = Tx$ , i.e.,  $y_i = \sum_{j=1}^m t_{ij} \cdot x_j$ . In view of this, a reasonable way to select a finite-parametric set is to pick a bounded symmetric convex set  $S_0$  with non-empty interior, and to use images  $TS_0$  of this set  $S_0$  under arbitrary linear transformations  $T$ .

If we start with a Euclidean unit ball  $S_0 = B \stackrel{\text{def}}{=} \left\{ x : \sum_{i=1}^n x_i^2 \leq 1 \right\}$ , we get the family of ellipsoids (see, e.g., [1–4, 11–14, 16]); if we start with a unit cube  $S_0 = C \stackrel{\text{def}}{=} \{x : |x_i| \leq 1 \text{ for all } i\}$ , we get the family of all boxes (plus the corresponding parallelepipeds); alternatively, we can also start with a symmetric convex polyhedron  $P$ .

**Which set  $S_0$  should we choose?** Once we pick a set  $S_0$ , we can (precisely) represent sets  $S$  of the type  $TS_0$ . If we start with such a set  $S$ , we enclose it into a set  $TS_0 = S$ , and then, if we want to enclose  $TS_0$  in a set  $\lambda \cdot S$  corresponding to the original  $S$ -based representations, we get the same original set  $S = TS_0$  back, with  $\lambda = 1$ .

For sets  $S$  which are different from  $TS_0$ , the  $S_0$ -based representation is only approximate. We start with a set  $S$ , and we enclose it in a set  $TS_0 \supseteq S$  for an appropriate linear transformation  $T$ . If we then try to enclose  $TS_0$  in a set of the type  $\lambda \cdot S$ , then we inevitably get  $\lambda > 1$ .



The smaller  $\lambda$ , the better the approximation. It is therefore reasonable, as a measure  $d(S_0, S)$  of accuracy of approximating  $S$  by  $S_0$ , to use the smallest  $\lambda$  corresponding to all possible  $T$ :

$$d(S_0, S) = \inf\{\lambda : \exists T (S \subseteq TS_0 \subseteq \lambda \cdot S)\}.$$

This quantity is known as a *Banach-Mazur distance* between the convex sets  $S$  and  $S_0$ ; see, e.g., [15, 17].

For each “standard” set  $S_0$ , we get different values  $A(S_0, S)$  for different sets  $S$ . As a measure of quality  $Q(S_0)$  of choosing  $S_0$ , it is reasonable to select the worst-case approximation accuracy

$$Q(S_0) \stackrel{\text{def}}{=} \sup_S d(S_0, S),$$

where the supremum is taken over all possible bounded symmetric convex sets  $S$  with non-empty interior.

## 2. Main Results

**Main conclusion: ellipsoids are better than generic polyhedra.** According to the well-known John's Theorem [8, 15, 17], for the Euclidean unit ball  $B$ , we have  $d(B, S) \leq \sqrt{n}$  for all symmetric convex sets  $S$ . Thus, we have  $Q(B) \leq \sqrt{n}$ .

On the other hand, according to Gluskin's theorem [6, 15, 17], there exists a constant  $c > 0$  such that for each dimension  $n$ , there exist polyhedra  $P$  and  $P'$  for which  $d(P, P') \geq c \cdot n$  and for which, therefore,  $d(P) \geq c \cdot n$ . Moreover, if we take a convex hull  $P$  of  $2n$  points randomly selected from a unit Euclidean sphere, then, with high probability, we get  $Q(P) \geq c \cdot n$ . Since for large  $n$ , we have  $c \cdot n \gg \sqrt{n}$  and therefore,  $Q(B) \ll Q(P)$ , this shows that for large dimensions, ellipsoids are indeed better than generic polyhedra.

**Additional conclusion: ellipsoids are probably better than boxes.** A Euclidean unit ball  $B$  (corresponding to ellipsoids) and a unit cube  $C$  (corresponding to boxes) can be viewed as particular cases of unit balls  $B_p \stackrel{\text{def}}{=} \{x : \|x\|_p \leq 1\}$  in the  $\ell_p$ -metric  $\|x\|_p \stackrel{\text{def}}{=} \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$ :  $B$  is a unit ball in the  $\ell_2$ -metric while  $C$  is a unit ball in the  $\ell_\infty$ -metric:  $B = B_2$  and  $C = B_\infty$ . The exact values of  $d(B_p, B_q)$  are known only when both  $p$  and  $q$  are on the same side of 2; in this case,  $d(B_p, B_q) = n^{|1/p-1/q|}$ . In particular, for  $p = 1$  and  $q = 2$ , we get  $d(B_1, B_2) = \sqrt{n}$ .

These values have the property that when  $p < q$ , then  $d(B_p, B_q)$  strictly increases when  $p$  decreases or when  $q$  increases; in other words, the larger the difference between  $p$  and  $q$ , the larger the value  $d(B_p, B_q)$ . For values  $p$  and  $q$  on different sides of 2, this monotonicity does not hold for  $n = 2$ , since in this case,  $B_1$  (rhombus) and  $B_\infty$  (square) are linearly equivalent and thus,  $d(B_1, B_\infty) = 0$ . However, for  $n > 3$ , we do not have this anomaly and therefore, it is reasonable to conjecture that for  $n > 3$ , this monotonicity holds. Under this hypothesis,  $d(B_\infty, B_1) > d(B_2, B_1) = \sqrt{n}$ , and thus,  $Q(B_\infty) \geq d(B_\infty, B_1) > \sqrt{n}$ . Since  $Q(B_2) = \sqrt{n}$ , we therefore conclude that  $Q(B_2) < Q(B_\infty)$  and thus, ellipsoids are better than boxes.

*Comment.* These results are in line with a general result according to which, under certain conditions, ellipsoids are the best approximators [5, 9].

**Acknowledgments.** This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant on F-transforms from the Office of Naval Research.

## REFERENCES

1. Belforte G., Bona B. An improved parameter identification algorithm for signal with unknown-but-bounded errors. Proceeding of the 7th IFAC Symposium on Identification and Parameter Estimation, York, U.K., 1985.

2. Chernousko F.L. Estimation of the Phase Space of Dynamic Systems. Moscow : Nauka publ., 1988 (in Russian).
3. Chernousko F.L. State Estimation for Dynamic Systems. Boca Raton, FL : CRC Press, 1994.
4. Filippov A.F. Ellipsoidal estimates for a solution of a system of differential equations // Interval Computations. 1992. N. 2(4). P. 6–17.
5. Finkelstein A., Kosheleva O., Kreinovich V. Astrogeometry, error estimation, and other applications of set-valued analysis // ACM SIGNUM Newsletter. 1996. V. 31, N. 4. P. 3–25.
6. Gluskin E.D. The diameter of the Minkowski compactum is approximately equal to  $n$  // Functional Analysis and Its Applications. 1981. V. 15. P. 72–73.
7. Jaulin L., Kieffer M., Didrit O., Walter E. Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics. London : Springer-Verlag, 2001.
8. John F. Extremum problems with inequalities as subsidiary conditions. In: Studies and Essays Presented to R. Courant on his 60th Birthday. New York : Interscience Publishers, Inc., 1948. P. 187–204.
9. Li S., Ogura Y., Kreinovich V. Limit Theorems and Applications of Set Valued and Fuzzy Valued Random Variables. Dordrecht : Kluwer Academic Publishers, 2002.
10. Moore R.E., Kearfott R.B., Cloud M.J. Introduction to Interval Analysis. Philadelphia, Pennsylvania : SIAM Press, 2009.
11. Norton J.P. Identification and application of bounded parameter models. Proceeding of the 7th IFAC Symposium on Identification and Parameter Estimation. York, U.K., 1985.
12. Schweppe F.C. Recursive state estimation: unknown but bounded errors and system inputs // IEEE Transactions on Automatic Control. 1968. V. 13. P. 22.
13. Schweppe F.C. Uncertain Dynamic Systems. Englewood Cliffs, New jersey : Prentice Hall, 1973.
14. Soltanov S.T. Asymptotic of the function of the outer estimation ellipsoid for a linear singularly perturbed controlled system / Shary S.P., Shokin Yu.I. (eds.) Interval Analysis. Krasnoyarsk, Academy of Sciences Computing Center, Technical Report N. 17, 1990. P. 35–40 (in Russian).
15. Tomczak-Jaegermann N. Banach-Mazur Distance and Finite-Dimensional Operator Ideals. Pitman Monographs and Surveys in Pure and Applied Mathematics. V. 38. Harlow, UK : Longman Sci. Tech., 1989.
16. Utyubaev G.S. On the ellipsoid method for a system of linear differential equations / Shary S.P. (eds.) Interval Analysis. Krasnoyarsk, Academy of Sciences Computing Center, Technical Report N. 16, 1990. P. 29–32 (in Russian).
17. Vershynin R. Lectures in Geometric Functional Analysis. Ann Arbor, Michigan : University of Michigan, 2013. URL: <http://www-personal.umich.edu/~romanv/> (the date of circulation: 31.03.2013).

## ОПИСАНИЕ DDoS-АТАКИ С ПОМОЩЬЮ КАТАСТРОФЫ «СБОРКА»

**А.К. Гуц, Д.Н. Лавров**

Показано, что DDoS-атаки на компьютерные системы можно описать с помощью катастрофы «сборка».

*DDoS*-атаки — наиболее распространённая атака злоумышленников на компьютерный информационный ресурс. Существуют два способа добиться от сервера отказа в обслуживании (*Denial of Service*).

Первый способ позволяет остановить работу *всей* атакуемой компьютерной системы. Для этого злоумышленник посылает серверу-жертве данные или пакеты, которые она не ожидает, и это приводит либо к остановке системы, либо к её перезагрузке. В результате никто не сможет получить доступ к ресурсам. Атака хороша тем, что с помощью нескольких пакетов можно сделать систему неработоспособной.

Второй способ (*Flood*-атаки) состоит в том, чтобы добиться переполнения системы с помощью такого большого количества пакетов, которое невозможно обработать. Например, если система может обрабатывать только 10 пакетов в секунду, а злоумышленник отправляет к ней 20 пакетов в секунду, то остальные пользователи при попытке подключиться к системе получают отказ в обслуживании, поскольку все ресурсы заняты. При таких атаках значительно снижается производительность компьютерной системы или приложений. Очевидно, что при этом способе атаки наблюдается резкое возрастание входящего трафика.

Есть и третий способ атаки, при которой стараются добиться переполнения канала, т.е. резко снизить пропускную способность канала.

Целью нашей заметки является математическое описание второго способа *DDoS*-атаки.

Мы видим, что для этого способа, во-первых, важную роль играет *входящий трафик*. Трафик — это параметр  $\tau$ , характеризующий типичную ситуацию для функционирующей компьютерной системы, которая говорит, что, как правило, ежедневный трафик именно таков, и система способна с ним справиться с определённым запасом надёжности системы.

Увеличение трафика требует для его обработки увеличения свободных ресурсов системы.

Во-вторых, мы видим, что важным параметром стойкости, надёжности компьютерной системы является ее *производительность*  $p$ , выражающаяся как в скорости обработки входящих пакетов, так и количестве устанавливаемых соединений.

При получении сервером пакета данных происходит его обработка. Это требует времени и определённых ресурсов компьютерной системы. Если приходит новый пакет, а сервер занят приёмом или обработкой предыдущего или другого пакета, то вновь приходящий запрос-пакет *ставится в очередь*, занимая при этом часть ресурсов системы.

При *Flood*-атаках происходит исчерпание ресурсов, а точнее ресурсов процессора, памяти или каналов связи, сводящееся к следующим моментам:

- ограниченное количество соединений, находящихся в состоянии установки (соединения), которыми располагает система (при TCP SYN *Flood*- и TCP *Flood*-атаках направляется большое количество запросов на инициализацию TCP-соединения с потенциальной системой-жертвой). Добиваются того, что система не может устанавливать новые соединения,
- способность системы автоматически отвечать на отправленные ping-запросы (*ICMP Flood*-атаки, *Smurf*-атаки). Если запрос использует большие (64 кБ) ICMP-пакеты, то они подвергаются фрагментации. Большое количество фрагментированных пакетов, могут привести к зависанию атакуемой системы, расходуя свои ресурсы на сборку.
- снижение пропускной способности канала связи за счёт потока большого количества UDP-пакетов разного размера (*UDP Flood*-атаки). Происходит перегрузка канала связи, и сервер, работающий по протоколу TCP, перестаёт отвечать.

Таким образом, способность к нормальному функционированию определяется числом откликов на запросы.

Обозначим через  $x(t)$  число откликов на запросы в момент времени  $t$ .

Тогда

$$x(t + 1) = x(t) + f[x(t)] + \tau, \quad (1)$$

где  $f[x(t)]$  — результат работы системы по обработке запросов на момент  $t$ . В уравнении отражено требование, что больший трафик требует нарастания числа откликов на запросы.

Примем для простоты, что  $f[x(t)] = kx(t)$ , где  $k$  — величина, определяющая производительность системы

$$k = \{p - g[x(t)]\}, \quad (2)$$

сводящаяся к средней скорости обработки входящих пакетов  $p$  с учётом её падения или увеличения в зависимости от объёма занятых ресурсов: чем больше загружены ресурсы, тем меньше скорость обработки входящих пакетов.

Пакет  $x$ , стоящий в очереди, должен пройти через соединение (либо просто пройти по забитому каналу, как UDP-пакет) и после обработки, возможно, породить отклик для пославшего его компьютера. Иначе говоря, пакет участвует в процессе его обработки как минимум дважды. Поэтому мы это отразим путём принятия предположения, что  $g[x] = x^2$ .

Таким образом,  $g[x(t)] = [x(t)]^2$  и тогда

$$x(t+1) = x(t) + [(p - p_0) - x^2(t)]x(t) + (\tau - \tau_0), \quad (3)$$

где введены некоторые «типичные» характерные для данного сервера величины производительности  $p_0$  и трафика  $\tau_0$ . При переходе к непрерывному времени уравнение (3) сводится к уравнению

$$\frac{dx}{dt} = [(p - p_0) - x^2(t)]x(t) + (\tau - \tau_0)$$

или

$$\frac{dx}{dt} = -\frac{\partial}{\partial x}V(x, p, \tau), \quad (4)$$

где

$$V(x, p, \tau) = \frac{1}{4}x^4 - \frac{1}{2}(p - p_0)x^2 - (\tau - \tau_0)x. \quad (5)$$

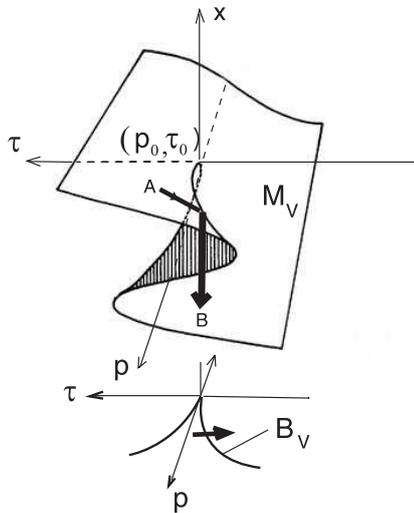


Рис. 1. Катастрофа сборки

Из вида выражения (5) видим, что сервер — это потенциальная динамическая система, потенциал которой описывается катастрофой «сборка» [1].

Естественно предположить, что в повседневных рутинных условиях сервер имеет в среднем одни и те же производительность  $p$  и трафик  $\tau$ . При этом число откликов в среднем является более или менее постоянным, т.е.  $x(t) = x_0 = const$ . В таком случае

$$\frac{dx}{dt} = 0$$

и, следовательно,  $x_0 = x_0(p, \tau)$  — это решение уравнения

$$\frac{\partial}{\partial x}V(x_0, p, \tau) = 0.$$

Такие решения называются состояниями *стационарного равновесия* системы. Сервер, таким образом, пребывает, как правило, в состоянии стационарного равновесия. Точки-равновесия  $(x_0, p, \tau)$  находятся в пространстве с осями  $x, p, \tau$  и началом  $(0, p_0, \tau_0)$  на поверхности  $M_V$  (рис.1). Из рисунка видно, что если

компьютерная система имела производительность  $p < p_0$ , т.е. не очень высокую, и трафик  $\tau > \tau_0$  и находилась в равновесии  $A$ , то при нарастании трафика (жирная стрелка на рисунке от  $A$  к  $B$ ) происходит скачкообразное обрушение такой характеристики, как количество откликов на запросы. Другими словами, происходит переход к равновесию «упавшего» сервера  $B$ .

#### ЛИТЕРАТУРА

1. Брёкер Т., Ландер Л. Дифференцируемые ростки и катастрофы. М. : Мир, 1977.

## **МОДЕЛЬ С НЕПРЕРЫВНЫМ ПРЕДСТАВЛЕНИЕМ ВРЕМЕНИ ДЛЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЙ С ГРУППИРОВКОЙ МАШИН ПО ТЕХНОЛОГИЯМ**

**Ю.В. Коваленко**

Рассматривается задача составления расписаний многопродуктового производства. Особенностью постановки является то, что каждый продукт имеет несколько технологий производства, при выполнении которых используется сразу несколько машин, работающих одновременно. Если машина переключается с одной технологии на другую, то необходимо выполнять переналадку. Построены модели частично целочисленного линейного программирования для задачи в общей постановке и для случая, когда длительности переналадки удовлетворяют неравенству треугольника. Для сравнения предложенных моделей проведены численные эксперименты на построенных случайным образом тестовых примерах.

### **Введение**

Большое практическое значение имеют задачи составления расписаний для производства, где в процессе выполнения операций на имеющемся множестве машин происходит получение одних веществ из других. Вещества подразделяются на сырье, промежуточные и окончательные продукты. Различают задачи с прерываниями и без прерываний. В задачах с прерываниями каждая операция может быть прервана и возобновлена позднее (см., например, [1, 4]), а в задачах без прерываний — не допускаются прерывания выполнения операций (см., например, [6, 10]). Продукты могут производиться либо непрерывно [14, 17], либо партиями [7, 13]. Кроме того, необходимо учитывать погрузку, хранение и транспортировку продуктов, переналадку машин и т. д.

Ряд работ (см., например, [4, 9, 11, 12, 16, 19]) отличается тем, что в них рассматриваются не отдельные операции, а технологии, представляющие собой набор операций, в результате действия которых может быть получен тот или иной продукт. Для выполнения технологии необходимо наличие некоторой совокупности машин, работающих одновременно или последовательно друг за

другом. Каждый продукт может быть произведён одной или более технологиями.

В настоящей статье рассматривается задача составления расписаний с группировкой машин по технологиям, возникающая, например, при производстве пластмасс и представляющая собой частный случай задачи, предложенной в [11]. Здесь в качестве операций рассматриваются химические реакции. Особенностью постановки является то, что каждый продукт имеет несколько технологий производства, при выполнении которых используется сразу несколько машин, работающих одновременно (в [4, 9, 12, 16] такие технологии называются многопроцессорными работами). Если машина переключается с одной технологии на другую, то необходимо выполнять переналадку. В соответствии с известной нотацией [3, 7, 16] рассматриваемая задача обозначается через  $P|\text{set}_i, sl_{uq}|C_{\max}$ , если прерывание выполнения технологий не допускается, и  $P|\text{set}_i, pmtn, sl_{uq}|C_{\max}$  — в противном случае.

В [3, 5, 9, 12, 16, 18] проводится анализ сложности задач составления расписаний с группировкой машин по технологиям, в которых длительности переналадок машин равны нулю. В задачах составления производственных расписаний [2, 7] рассматриваются технологии, задействующие при своём выполнении только одну машину, но уже при наличии переналадок. В [11] технологии рассматриваются при обсуждении постановки задачи, однако в модели они в явном виде не представлены. Такой подход приводит к моделям частично целочисленного линейного программирования достаточно большой размерности, для решения которых применяется метод декомпозиции [11, 13, 14].

В настоящей работе предложены модели частично целочисленного линейного программирования (ЧЦЛП) для задач теории расписаний  $P|\text{set}_i, sl_{uq}|C_{\max}$  и  $P|\text{set}_i, pmtn, sl_{uq}|C_{\max}$  в общей постановке и для случая, когда длительности переналадки удовлетворяют неравенству треугольника. Построенные модели учитывают технологии в явном виде и имеют меньшее число переменных и ограничений, чем модель [11] в применении к исследуемой задаче.

Модель, сформулированная для случая выполненного неравенства треугольника, является более предпочтительной при использовании методов оптимизации, основанных на ЛП-релаксации, что подтверждается численными экспериментами на сгенерированных случайным образом тестовых примерах.

## 1. Постановка задачи

Рассмотрим предприятие, выпускающее  $k$  различных продуктов. Требуемый объем производства продукта  $i$ ,  $i = 1, \dots, k$  обозначим через  $V_i \in \mathbb{R}^+$  (здесь и далее  $\mathbb{R}^+$  — множество положительных вещественных чисел). Пусть  $m$  — число машин, которые могут использоваться при выпуске продукции.

Для каждого продукта  $i$ ,  $i = 1, \dots, k$  указана одна или более технологий его производства. Пусть  $U$  — множество технологий, каждая из которых характеризуется набором одновременно занимаемых машин  $M_u \subseteq \{1, \dots, m\}$ ,  $u \in U$ , т.е. если производится продукт  $i$  по технологии  $u$ , то одновременно задействованы все машины, относящиеся к данной технологии. В любой момент каждая маши-

на не может быть задействована более чем в одной технологии. Обозначим  $|U|$  через  $d$ .

Пусть  $U_i \subseteq U$  — множество технологий по производству продукта  $i$ ,  $i = 1, \dots, k$ , для каждой из которых задан объем  $a_u \in \mathbb{R}^+$  выпуска данного продукта в единицу времени,  $u \in U_i$ . Предполагается, что для выпуска продукта  $i$  может быть использовано несколько технологий из множества  $U_i$ ,  $i = 1, \dots, k$ . В соответствии с терминологией [8, 16] допускается перемещение (migration) по технологиям при выпуске продукта.

Для машины  $l$  заданы длительности переналадки этой машины с технологии  $u$  на технологию  $q$ , обозначенные через  $s_{luq} \in \mathbb{R}_+$  для всех  $u, q \in K_l$ , где  $K_l = \{u \in U : l \in M_u\}$  — множество технологий, использующих машину  $l$ ,  $l = 1, \dots, m$  (здесь  $\mathbb{R}_+$  — множество неотрицательных вещественных чисел).

Для каждого продукта  $i$ ,  $i = 1, \dots, k$  необходимо определить, какие технологии  $u \in U_i$  будут использоваться для его производства, и для выбранных технологий составить расписание их выполнения с учётом переналадок машин и невозможности одновременного использования одной машины в различных технологиях таким образом, чтобы общее время окончания производства всех продуктов  $C_{\max}$  в объёмах  $V_1, \dots, V_k$  было минимально. Задача рассматривается в двух вариантах: с возможностью прерываний выполнения технологий ( $P|set_i, pmtn, s_{luq}|C_{\max}$ ) и без неё ( $P|set_i, s_{luq}|C_{\max}$ ). Ввиду возможности перемещения по технологиям при выпуске продукта в центральное поле обозначения задач следовало бы добавить символ "var" [8], но для компактности изложения этот символ опускается.

На практике достаточно часто встречаются задачи составления производственных расписаний, где длительности переналадки удовлетворяют неравенству треугольника:

$$s_{luq} + s_{lqp} \geq s_{lup}, \quad l = 1, \dots, m, \quad u, q, p \in K_l. \quad (1)$$

Обозначим этот частный случай через  $P|set_i, pmtn, \Delta s_{luq}|C_{\max}$ , если допускается прерывание выполнения технологий, и  $P|set_i, \Delta s_{luq}|C_{\max}$ , если не допускается. Указанные задачи являются NP-трудными в сильном смысле, так как в частном случае при  $m = 1$  к ним сводится метрическая задача о кратчайшем гамильтоновом пути, являющаяся NP-трудной в сильном смысле [15].

## 2. Модель частично целочисленного линейного программирования для общего случая

Известно множество подходов к формулированию задач построения производственных расписаний в виде моделей ЧЦЛП с учётом сложных связей между технологиями, продуктами и машинами [2, 11, 13, 14, 19]. Учитывая структуру рассматриваемой задачи, можно предложить модель, основанную на тех же принципах, что и в [2, 11, 13].

Определим понятие точки событий, аналогичное введённому в [13]. *Точка событий* — это группа переменных задачи, которые задают некоторый набор

технологий и моменты начала и окончания выполнения технологий из этого набора. В одной точке событий каждая машина может быть задействована не более чем в одной технологии. Множество точек событий обозначим через  $N = \{1, \dots, n_{\max}\}$ , где параметр  $n_{\max}$  выбирается достаточно большим на основе априорных оценок или вычислительных экспериментов.

Основными переменными для построения модели будут булевы переменные  $w_{un}$  такие, что  $w_{un} = 1$ , если технология  $u$  выполняется в точке событий  $n$ , и  $w_{un} = 0$  иначе. Кроме того, введём вещественные переменные, которые в случае, если технология  $u$  выполняется в точке событий  $n$ , имеют следующий смысл:  $T_{un}^s$  — время начала выполнения технологии  $u$  в точке событий  $n$ ,  $T_{un}^f$  — время завершения выполнения технологии  $u$  в точке событий  $n$ . Переменная  $C_{\max}$  соответствует моменту завершения производства всех продуктов.

Введем следующие обозначения:

$I$  — множество продуктов,  $|I| = k$ ;

$M$  — множество машин,  $|M| = m$ ;

$H = \sum_{i \in I} \max_{u \in U_i} \left\{ \frac{V_i}{a_u} \right\} + (k-1) \cdot \max_{l \in M, u, q \in K_l} \{s_{luq}\}$  — оценка сверху длины расписания.

Времени  $H$  заведомо достаточно для выпуска всех продуктов.

Тогда модель ЧЦЛП для задачи  $P|set_i, pmtn, s_{luq}|C_{\max}$  может быть записана следующим образом:

$$C_{\max} \rightarrow \min, \tag{2}$$

$$T_{un}^f \leq C_{\max}, \quad u \in U, \quad n \in N, \tag{3}$$

$$\sum_{u \in K_l} w_{un} \leq 1, \quad l \in M, \quad n \in N, \tag{4}$$

$$T_{un}^s \geq T_{q\tilde{n}}^f + s_{lqu} - H \cdot (2 - w_{un} - w_{q\tilde{n}} + \sum_{q' \in K_l} \sum_{\tilde{n} < n' < n} w_{q'n'}), \tag{5}$$

$$l \in M, \quad u, q \in K_l, \quad n, \tilde{n} \in N, \quad n \neq 1, \quad \tilde{n} < n,$$

$$T_{un}^f \geq T_{un}^s, \quad u \in U, \quad n \in N, \tag{6}$$

$$T_{un}^f - T_{un}^s \leq w_{un} \cdot \max_{q \in U_i} \left\{ \frac{V_i}{a_q} \right\}, \quad i \in I, \quad u \in U_i, \quad n \in N, \tag{7}$$

$$\sum_{n \in N} \sum_{u \in U_i} a_u \cdot (T_{un}^f - T_{un}^s) \geq V_i, \quad i \in I, \tag{8}$$

$$T_{un}^s \geq 0, \quad u \in U, \quad n \in N, \tag{9}$$

$$w_{un} \in \{0, 1\}, \quad u \in U, \quad n \in N. \tag{10}$$

Целевая функция (2) и неравенство (3) задают критерий минимизации момента окончания производства всех продуктов. Ограничение (4) выражает тот факт, что в каждой точке событий машина  $l$  используется не более чем в одной технологии, ограничение (5) — что время начала технологии  $u$  на машине  $l$  должно быть не меньше, чем время окончания предыдущей технологии на той же машине плюс длительность переналадки. Условие (6) гарантирует неотрицательность длительности технологий. Если технология  $u$  в точке событий  $n$  не

выполняется, то её длительность должна быть равна нулю, что обеспечивается неравенством (7). Ограничение (8) гарантирует выпуск продукции в заданном объёме. Условия (9) – (10) описывают область определения переменных.

Модель ЧЦЛП для задачи  $P|set_i, s_{luq}|C_{\max}$  может быть получена путём добавления к ограничениям (2) – (10) неравенства

$$\sum_{n \in N} w_{un} \leq 1, \quad u \in U, \quad (11)$$

гарантирующего непрерывность выполнения каждой технологии.

### 3. Модель частично целочисленного линейного программирования для случая выполненного неравенства треугольника

На практике достаточно часто возникают задачи составления производственных расписаний, в которых длительности переналадки удовлетворяют неравенству треугольника (1). Поэтому имеет смысл сформулировать модели ЧЦЛП для задач  $P|set_i, pmtn, \Delta s_{luq}|C_{\max}$  и  $P|set_i, \Delta s_{luq}|C_{\max}$ . Предлагаемые модели представляются более предпочтительными, так как в них удаётся исключить неравенство (5), которое усложняет процесс поиска оптимального решения для методов оптимизации, основанных на ЛП-релаксации.

С использованием введённых ранее обозначений модель ЧЦЛП для задачи  $P|set_i, pmtn, \Delta s_{luq}|C_{\max}$  может быть записана следующим образом:

$$C_{\max} \rightarrow \min, \quad (12)$$

$$T_{un}^f \leq C_{\max}, \quad u \in U, \quad n \in N, \quad (13)$$

$$\sum_{u \in K_i} w_{un} \leq 1, \quad l \in M, \quad n \in N, \quad (14)$$

$$T_{u,n+1}^s \geq T_{un}^f, \quad u \in U, \quad n \in N, \quad n \neq n_{\max}, \quad (15)$$

$$T_{u,n+1}^s \geq T_{qn}^f + s_{lqu} \cdot w_{u,n+1} - H \cdot (1 - w_{u,n+1}), \quad (16)$$

$$l \in M, \quad u, q \in K_l, \quad u \neq q, \quad n \in N, \quad n \neq n_{\max},$$

$$T_{un}^s \geq -H \cdot (1 - w_{un}), \quad u \in U, \quad n \in N, \quad (17)$$

$$T_{un}^f \geq T_{un}^s, \quad u \in U, \quad n \in N, \quad (18)$$

$$T_{un}^f - T_{un}^s \leq w_{un} \cdot \max_{q \in U_i} \left\{ \frac{V_i}{a_q} \right\}, \quad i \in I, \quad u \in U_i, \quad n \in N, \quad (19)$$

$$\sum_{n \in N} \sum_{u \in U_i} a_u \cdot (T_{un}^f - T_{un}^s) \geq V_i, \quad i \in I, \quad (20)$$

$$w_{un} \in \{0, 1\}, \quad u \in U, \quad n \in N. \quad (21)$$

Ограничения (12) – (14) и (18) – (21) имеют тот же смысл, что и в предыдущей модели. Условие (15) выражает то, что время начала выполнения технологии  $u$  в точке событий  $n+1$  должно быть не меньше, чем время окончания её выполнения в точке событий  $n$ . Неравенство (16) говорит о том, что время начала технологии  $u$  на машине  $l$  должно быть не меньше, чем время окончания предыдущей технологии на той же машине плюс длительность переналадки. Однако, если технология  $u$  является первой на машине  $l$ , то длительность переналадки на неё учитывать не нужно. Это обеспечивается благодаря тому, что переменные  $T_{qn}^s$  во всех предшествующих точках событий могут принимать отрицательные значения. Если же технология  $u$  имеет место в точке событий  $n$ , то время начала её выполнения должно быть неотрицательным, что обеспечивается условием (17).

Необходимо отметить, что ограничение (16) будет гарантировать получение оптимального решения задачи  $P|set_i, pmtn, s_{luq}|C_{max}$ , только если длительности переналадки удовлетворяют неравенству треугольника.

Модель ЧЦЛП для задачи  $P|set_i, \Delta s_{luq}|C_{max}$  может быть получена путём добавления к ограничениям (12) – (21) неравенства (11).

#### 4. Вычислительный эксперимент

Для оценки применимости моделей, предложенных для случая выполненного неравенства треугольника, проведён вычислительный эксперимент на построенных случайным образом задачах трёх серий  $S1$ ,  $S2$  и  $S3$ .

Модели (2) – (10) и (12) – (21) ((2) – (11) и (11) – (21)) были записаны в системе моделирования GAMS 23.2 и задача  $P|set_i, pmtn, \Delta s_{luq}|C_{max}$  ( $P|set_i, \Delta s_{luq}|C_{max}$ ) решалась с помощью универсального пакета CPLEX 12.1. При этом использовался метод ветвей и границ с отсечениями, настройки которого были выбраны по умолчанию. Тестирование проводилось на ЭВМ Intel Core2 Duo CPU E7200 2.54 ГГц, оперативная память 2 Гб.

При генерации тестовых задач для каждой технологии  $u \in U$  число машин  $|M_u| \in \{1, \dots, m\}$  выбиралось случайно, а затем машины назначались на данную технологию равновероятно без повторений. Числовые значения для всех тестовых задач генерировались случайным образом с равномерным распределением из следующих множеств:  $V_i \in [1, V_{max}]$ ;  $|U_i| \in \{1, \dots, U_{max}\}$ ;  $a_u \in [1, \frac{V_i}{2}]$ , где  $i$  такое, что  $u \in U_i$ ;  $s_{luq} \in [0, s_{max}]$ . В табл. 1 приведены выбранные значения параметров для каждой серии.

Таблица 1. Параметры серий

серия	число задач	$k$	$m$	$U_{max}$	$V_{max}$	$s_{max}$	$n_{max}$
$S1$	10	4	4	3	10	5	5
$S2$	10	5	7	5	12	7	6
$S3$	10	8	10	5	15	9	8

Для проведения вычислительного эксперимента было установлено максимальное время, равное 7200 сек. CPLEX может решать за указанное время один тестовый пример.

В табл. 2, 3 и 4 представлены результаты вычислительного эксперимента для задачи  $P|set_i, pmtn, \Delta s_{luq}|C_{max}$ . Здесь используются следующие обозначения:

$N_{var}$  — число переменных в моделях (2) – (10) и (12) – (21);

$N_{eqv}$  — число уравнений в модели (2) – (10);

$N_{eqv\Delta}$  — число уравнений в модели (12) – (21);

$C$  — значение целевой функции, полученное при решении задачи (2) – (10) пакетом CPLEX;

$C_{\Delta}$  — значение целевой функции, полученное при решении задачи (12) – (21) пакетом CPLEX;

$t$  — время работы CPLEX в сек. при решении задачи (2) – (10);

$t_{\Delta}$  — время работы CPLEX в сек. при решении задачи (12) – (21).

Таблица 2. Сравнение моделей на задачах серии S1

задача	$N_{var}$	$N_{eqv}$	$N_{eqv\Delta}$	$C$	$C_{\Delta}$	$t$	$t_{\Delta}$
1	121	834	432	15.5691	15.5691	39.14	16.52
2	151	1434	688	15.4431	15.4431	48.37	11.08
3	121	1054	504	17.5173	17.5173	17.26	6.96
4	106	589	328	11.1378	11.1378	3.39	0.79
5	121	924	464	16.2939	16.2939	6.73	2.88
6	91	444	256	13.4058	13.4058	1.44	0.53
7	106	359	248	7.8721	7.8721	0.7	0.33
8	121	974	488	20.3919	20.3919	9.45	4.57
9	106	649	344	11.8913	11.8913	7.61	3.19
10	121	854	440	21.7056	21.7056	54.2	26.47

Таблица 3. Сравнение моделей на задачах серии S2

задача	$N_{var}$	$N_{eqv}$	$N_{eqv\Delta}$	$C$	$C_{\Delta}$	$t$	$t_{\Delta}$
1	235	3656	1374	24.6338	24.6338	420	115.9
2	235	4541	1624	33.194	33.194	693.5	43.5
3	253	5654	2003	41.8958	41.8958	400	180.4
4	325	7826	2769	14.8397	14.8397	720.5	321.3
5	235	5441	1904	37.3962	37.3962	69.3	11.1
6	343	9479	3318	20.8407	20.8407	907.1	434.3
7	343	7409	2668	16.2408	16.2408	520.2	209.8
8	271	3917	1482	13.6062	13.6062	1111	494
9	235	3881	1434	34.9472	34.9472	194.9	43.7
10	199	2855	1066	48.6371	48.6371	43.8	20.9

Таблица 4. Сравнение моделей на задачах серии S3

задача	$N_{\text{var}}$	$N_{\text{eqv}}$	$N_{\text{eqv}\Delta}$	$C$	$C_{\Delta}$	$t$	$t_{\Delta}$
1	505	16048	4281	21.985	21.985	4327	3456
2	649	36912	9415	17.3601	17.3601	6257	2199
3	577	30120	7688	26.4992	26.4992	6887	4398
4	625	40948	10356	31.5768	31.5768	5434	5199
5	529	32508	8198	41.6226*	36.5283	7200	6589
6	601	35856	9085	29.2432*	25.8131	7200	3199
7	481	23752	6048	30.1493	30.1493	7200	2245
8	529	25844	6616	40.488*	39.6324	7200	5321
9	505	22096	5695	28.3952*	27.8647	7200	4567
10	529	25676	6574	31.7315	31.7315	7200	3199

Из табл. 2, 3 и 4 видно, что модель (12) – (21) содержит меньшее число ограничений, чем модель (2) – (10). Кроме того, в среднем на сериях S1 и S2 пакет CPLEX решает тестовые примеры, записанные в модели (12) – (21), более чем в два раза быстрее, чем при их записи в модели (2) – (10). На серии S3 при решении пакетом CPLEX задач 5 – 10, записанных в модели (2) – (10), за установленное время удается найти только допустимые решения, которые не всегда являются оптимальными (отмечены '\*'). Аналогичные результаты имеют место для задачи  $P|set_i, \Delta s_{luq}|C_{\max}$ .

Таким образом, если в задаче составления расписаний с группировкой машин по технологиям длительности переналадки удовлетворяют неравенству треугольника, то при её решении лучше использовать модель (12) – (21) при условии, что допускаются прерывания выполнения технологий, и модель (11) – (21) – в противном случае, так как данные модели имеют меньшее число ограничений и являются менее сложными для методов оптимизации, основанных на ЛП - релаксации (в том числе для пакета CPLEX).

## 5. Заключение

Рассмотрена задача составления расписаний многопродуктового производства. Особенностью данной задачи является то, что каждый продукт может производиться по нескольким технологиям, каждая из которых характеризуется набором одновременно занимаемых машин. Построены модели ЧЦЛП для задачи в общей постановке и для случая, когда длительности переналадки удовлетворяют неравенству треугольника. С помощью вычислительного эксперимента показано, что вторая модель является более предпочтительной для методов оптимизации, основанных на ЛП-релаксации.

Автор благодарит А.В. Еремеева за предложенную постановку задачи.

## ЛИТЕРАТУРА

1. Баптист Ф., Карлье Ж., Кононов А.В., Керан М., Севастьянов С.В., Свириденко М. Структурные свойства оптимальных расписаний с прерываниями операций // Дискрет. анализ и исслед. операций. 2009. Т. 16, № 1. С. 3–36.
2. Борисовский П.А. Генетический алгоритм для одной задачи составления производственного расписания с переналадками // Тр. XIV Байкальской международной школы-семинара «Методы оптимизации и их приложения». Иркутск : ИСЭМ СО РАН, 2008. Т. 4. С. 166–173.
3. Bianco L., Blazewicz J., Dell’Ohno P., Drozdowski M. Scheduling multiprocessor tasks on a dynamic configuration of dedicated processors // Ann. Oper. Res. 1995. V. 58. P. 493–517.
4. Bianco L., Blazewicz J., Dell’Ohno P., Drozdowski M. Scheduling preemptive multiprocessor tasks on dedicated processors // Performance Evaluation. 1994. V. 20. P. 361–371.
5. Blazewicz J., Dell’Ohno P., Drozdowski M., Speranza M.G. Scheduling multiprocessor tasks on three dedicated processors // Information Processing Letters. 1992. V. 41. P. 275–280. Corrigendum: V. 49. P. 269–270.
6. Bianco L., Dell’Ohno P., Speranza M.G. Nonpreemptive scheduling on independent tasks with prespecified processor allocations // Naval Research Logistics Quarterly. 1994. V. 41. P. 959–971.
7. Dolgui A., Ereemeev A.V., Kovalyov M.Y. Multi-product lot-sizing and scheduling on unrelated parallel machines: research report N. 2007-500-011. Saint-Etienne : Ecole des Mines de Saint-Etienne, 2007. 15 p.
8. Drozdowski M. Scheduling for parallel processing. London : Springer-Verl., 2009. 386 p.
9. Drozdowski M. Scheduling multiprocessor tasks – An overview // Eur. J. Oper. Res. 1996. V. 94. P. 215–230.
10. Du J., Leung J.Y-T. Complexity of scheduling parallel task systems // SIAM J. Discrete Math. 1989. V. 2, N. 4. P. 472–478.
11. Floudas C.A., Kallrath J., Pitz H.J., Shaik M.A. Production scheduling of a large-scale industrial continuous plant: short-term and medium-term scheduling // Comp. Chem. Engng. 2009. V. 33. P. 670–686.
12. Hoogeveen J.A., van de Velde S.L., Veltman B. Complexity of scheduling multiprocessor tasks with prespecified processors allocations // Discrete Appl. Math. 1994. V. 55. P. 259–272.
13. Ierapetritou M.G, Floudas C.A. Effective continuous-time formulation for short-term scheduling: I. multipurpose batch process // Ind. Eng. Chem. Res. 1998. V. 37. P. 4341–4359.
14. Ierapetritou M.G, Floudas C.A. Effective continuous-time formulation for short-term scheduling: II. continuous and semi-continuous processes // Ind. Eng. Chem. Res. 1998. V. 37. P. 4360–4374.
15. Itai A., Papadimitriou C.H., Szwarcfiter J.L. Hamilton paths in grid graphs // SIAM J. Comput. 1982. V. 11, N. 4. P. 676 – 686.
16. Jansen K., Porkolab L. Preemptive scheduling with dedicated processors: applications of fractional graph coloring // Journ. Scheduling. 2004. V. 7. P. 35–48.

17. Kondili E., Pantelides C.C., Shan N. Production planning for the rational use of energy in multiproduct continuous plants // *Comp. Chem. Engng.* 1993. V. 17. P. 123–136.
18. Kubale M. Preemptive versus nonpreemptive scheduling of biprocessor tasks on dedicated processors // *Eur. J. Oper. Res.* 1996. V. 94. P. 242–251.
19. Lin X., Floudas C.A., Modi S., Juhasz N.M. Continuous-time optimization approach for medium-range production scheduling of a multiproduct batch plant // *Ind. Eng. Chem. Res.* 2002. V. 41. P. 3884–3906.

## КОНЦЕПЦИЯ ПРОЕКТА РАСШИРЯЕМОГО ОБУЧАЮЩЕГО ПРОГРАММНОГО СРЕДСТВА

С.В. Гусс

В работе даётся описание концепции проекта расширяемого сетевого обучающего программного средства. Предлагаются параметры для управления обучающей системой. Представлены общие элементы.

### Введение

Предлагаемый общий проект представляет собой базу для конкретных рабочих проектов обучающих программных средств. В проекте есть *общая основа*, которая имеет *точки расширения*. Общая основа — фиксированная часть. Точки расширения предоставляют доступ к местам фиксированной части, которые требуют конфигурации. Фиксированная часть и инструменты для работы с точками расширения создаются в рамках *системы элементов повторного использования и автоматизации проектирования*. Это инфраструктура *расширяемой обучающей системы*. Обучающая система является программной базой для запуска модулей игровых систем (реализующих лингвистические игры), разработанных в ходе конкретных *проектов компонентов игровых обучающих программных средств*.

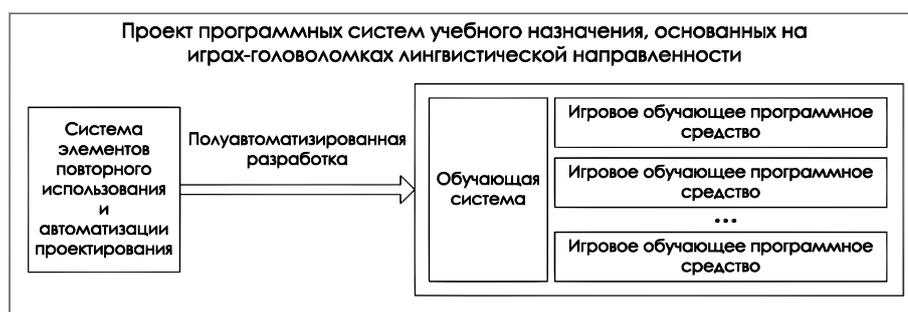


Рис. 1. Структура проекта

## 1. Процессы

**Образование** — процесс передачи знаний и методов их получения от *учителя* *ученику* в рамках образовательной стратегии одной из форм общественной организации, начиная от государства и заканчивая объединениями групп по интересам и заинтересованными в этом индивидуумами. **Самообразование** — образование, получаемое индивидуумом или группой индивидуумов, выступающих в роли *учеников*, в процессе которого значительная доля знаний осваивается без участия *учителя*, суть которого может сводиться к составлению и выдаче заданий (с последующей проверкой их выполнения) и рекомендаций *ученикам*; функции такого *учителя* могут быть реализованы техническими средствами. **Автоматизация образования** — совершенствование процесса образования, направленное на переложение рутинной работы участников процесса на технические средства и способствование этими средствами творческому акту *учеников* и *учителей*.

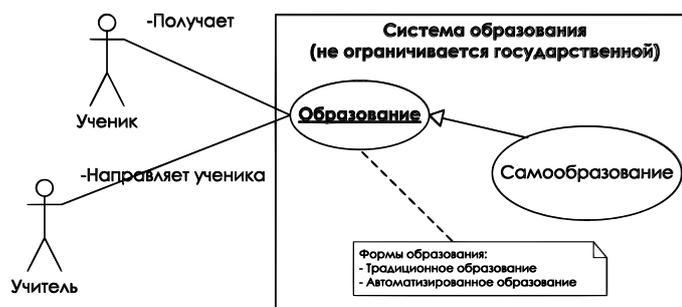


Рис. 2. Система образования

## 2. Обучающая система

**Обучающая система** — система, состоящая из набора взаимосвязанных подсистем, обеспечивающих работу всей системы для удовлетворения потребностей заинтересованных в *образовании* и *самообразовании* лиц. **Пользователь обучающей системы** — лицо, взаимодействующее с *обучающей системой* способами, предусмотренными её организаторами или создателями. **Учитель** — субъект процесса *образования*, выступающий в роли передающего знания. **Ученик** — субъект процесса *образования*, выступающий в роли получающего знания. **Администратор** — субъект процесса *образования*, ответственный за функционирование и поддержание работоспособности *обучающей системы*, в обязанности которого входит её настройка, конфигурирование и проведение испытаний.

*Примечание: процесс администрирования может быть автоматизирован (проверка целостности системы и данных, наличия необходимых системе компонентов перед запуском обучающей системы, восстановление данных и компонентов системы).*

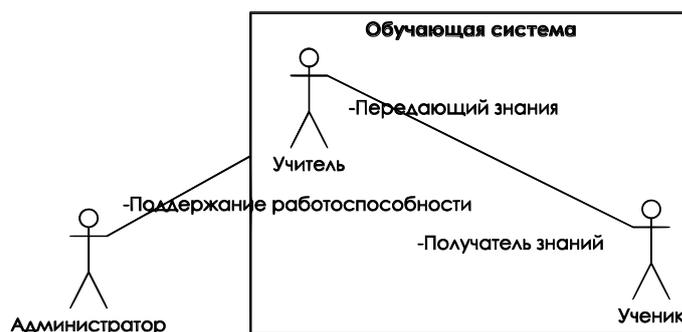


Рис. 3. Обучающая система



Рис. 4. Обучающая система (подсистемы)

### 3. Подсистема «Обучение»

**Подсистема «Обучение»** — подсистема поддержки учебного процесса, основная задача которой — организация учебных и проверочных занятий. **Учебный процесс** — этап процесса образования, характеризующийся активной деятельностью участников процесса обучения, в рамках которого происходит взаимодействие между ними и подсистемой «Обучение». **Учебное занятие** — часть учебного процесса, в рамках которого происходит передача знаний от учителя ученику посредством подсистемы «Обучение». **Проверочное занятие** — часть учебного процесса, в рамках которого происходит проверка знаний, полученных учениками в ходе учебных занятий. **Задача** — задания, выданные ученику и подлежащие решению в рамках занятия (проверочного или учебного). **Задание** — часть задачи, в рамках которой необходимо выполнить ряд шагов. **Ход** — действие, совершённое для выполнения определённого шага в процессе решения задачи.

*Примечание:* учебный процесс может состоять из нескольких учебных и проверочных занятий, которые предусмотрены учебным планом. В рамках занятий могут выдаваться различные задачи, способствующие закреплению полученных в рамках занятия знаний (учебное занятие) или их проверке (проверочное занятие).

*Примечание:* явным образом учитель может отсутствовать во время за-

нятий, в таком случае функции учителя реализуются техническими средствами.

*Уточняющий пример:* вопрос — шаг, ответ — совершённый ход. Лингвистические игры, разрабатываемые в рамках проекта, в большинстве своём, — игры типа «вопрос-ответ». Во взаимодействии ученика с системой это проявляется следующим образом. Система задаёт вопрос, ученик отвечает на него. Ответ ученика вызывает соответствующую реакцию системы.

**Общий диалог** возможен между всеми участниками процесса обучения. **Групповой диалог** возможен между учебной группой участников. **Приватный диалог** возможен между отдельными участниками процесса обучения. **Действующие субъекты** — одушевлённые и неодушевлённые объекты, чьё участие может прослеживаться в рамках подсистемы «Обучение». **Участники процесса обучения** — действующие субъекты, деятельность которых проявляется в рамках учебных и проверочных занятий.

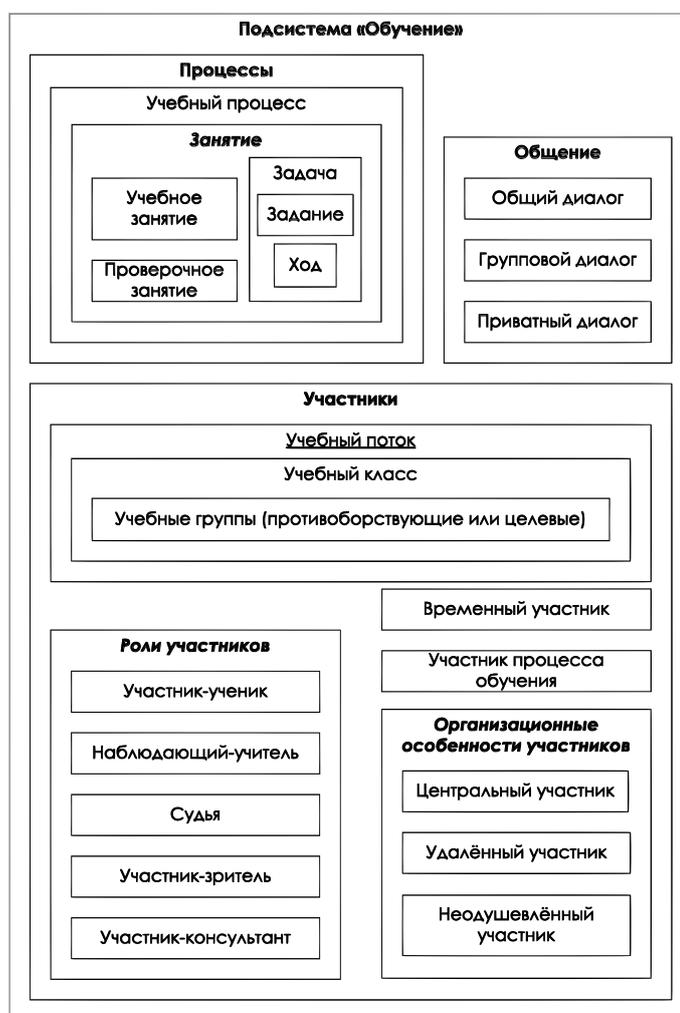


Рис. 5. Подсистема «Обучение»

**Временный участник** (не зарегистрированный в системе, либо не закреплённый за определённым учебным классом, но включённый в учебный поток) — действующий субъект, чья деятельность в рамках подсистемы «Обучение» носит несистемный характер или находится за рамками учебного плана. **Учебный поток** — набор участников процесса обучения, объединённых для достижения общей цели в рамках учебного процесса. **Учебный класс** — набор участников процесса обучения, объединённых для достижения общей цели в рамках занятий (учебных или проверочных). **Учебная группа** — набор участников процесса обучения, объединённых для достижения общей цели в рамках задачи. **Противоборствующие группы** — группы, между которыми идёт соревновательный процесс. Каждой такой группе выдаётся одинаковая задача. Результаты решения задачи сравниваются. Выявляется группа-победитель. **Целевые группы** — группы, на которые делятся участники процесса обучения в целях совместного решения задачи.

*Примечание: разным целевым группам могут выдаваться разные задачи. Если целевым группам выданы одинаковые задачи, их можно трансформировать в противоборствующие группы для сравнения результатов.*

**Участник-ученик** — субъект, чья активность подлежит анализу и оценке. **Наблюдающий-учитель** — субъект, чья деятельность связана с наблюдением за участником-учеником, предусматривающая оперативное вмешательство в случае необходимости в учебное или проверочное занятие. Обладает *судейским чином*. За ним всегда остаётся последнее слово в решении организационных и учебных вопросов в рамках подсистемы «Обучение». **Судья** — неодушевлённый объект, реализуемый техническими средствами, в чьи обязанности входит анализ учебного и проверочного занятия, а именно проверка действий участников на соответствие установленным правилам и принятие соответствующих ответных решений. Постоянный участник учебных и проверочных занятий. Для модификации логики его работы или её замещения необходимо обладать *судейским чином*. **Участник-зритель** — участник процесса обучения, активность которого связана с наблюдением за учебными и проверочными занятиями. **Участник-консультант** — участник процесса обучения, активность которого связана с наблюдением за участниками и предоставление им, в случае необходимости, вспомогательной информации.

*Примечание: участник-зритель может общаться в режиме общего диалога.*

*Примечание: участник-консультант может общаться в режиме общего диалога, приватного и группового.*

**Центральный участник** — участник процесса обучения, без присутствия которого не возможно учебное или проверочное занятие. **Удалённый участник** — участник процесса обучения, подключающийся к остальным участникам посредством установления связи с центральным участником. **Неодушевлённый участник** — участник процесса обучения, функции которого выпол-

няются с помощью технических средств.

*Примечание:* центральным участником может быть только один участник, выступающий в роли участника-ученика либо наблюдающего-учителя.

*Примечание:* удалённый участник может стать центральным участником в случае наделения его данной организационной особенностью текущим центральным участником.

#### 4. Подсистема «Управление пользователями»

**Подсистема «Управление пользователями»** — подсистема поддержки учебного процесса, основные задачи которой — регистрация участников и управление ими. **Регистрация участников** — запись информации в системе об участнике процесса обучения, создание *учётной записи*. **Анализ учётной записи** — просмотр, модификация, внесение изменений в *учётную запись* участника, оценивание результатов деятельности участника, не поддающейся оценке техническими средствами (например, в случае проверки развёрнутых ответов на вопросы в тестовом режиме). **Учётная запись** — идентификационные данные участника, включающие его *профиль* и предоставляющие доступ к *протоколу участника*. **Профиль** — информация об *участнике процесса обучения*, достаточная для представления о нём другим участникам. **Протокол участника** — информация о деятельности участника в рамках обучающей системы и предпочтительных настройках системы.

*Примечание:* деятельность участника (работа на занятиях, выполненные задачи, модификация профиля).

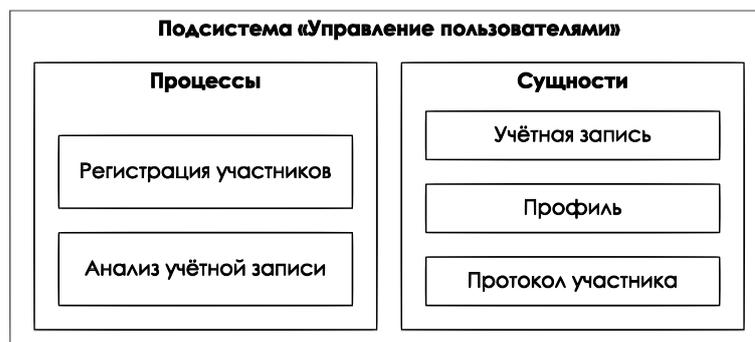


Рис. 6. Подсистема «Управление пользователями»

**Протокол участника. Деятельность.** Указываются моменты пребывания в системе: вид обучения, режим обучения, занятия, выполненные задачи, роль в рамках занятия, организационная особенность. **Временные характеристики.** Указывается время работы с системой и время, потраченное на занятие. Статистика решения задачи. **Профиль участника.** Указываются Ф.И.О., псевдоним (то, как пользователь желает, чтобы к нему обращались в системе), дата регистрации, сведения о себе.

## 5. Подсистема «Учебный материал»

**Подсистема «Учебный материал»** — подсистема поддержки *учебного процесса*, ответственная за хранение материала, необходимого для передачи знаний в рамках *занятий*. **Учебный материал** — информация, используемая в рамках *занятий* для передачи знаний или проверки качества её усвоения (это может быть раздаточный материал или конкретные задания).

**Виды материала.** *Раздаточный материал.* Текст. Иллюстрации. Видео. *Задачи.* Материал для генерации. Набор необходимой для автоматической генерации задачи информации, например словарь предметной области. Составленная задача.

**Операции.** Внесение материала. Просмотр материала. Редактирование материала.

## 6. Подсистема «Безопасность»

**Подсистема «Безопасность»** — подсистема поддержки *учебного процесса*, ответственная за вопросы аутентификации, секретности, целостности и строгого выполнения обязательств. **Аутентификация** — процесс проверки соответствия пользователя системы своей *учётной записи*. **Секретность** — защита информации (учебного материала, регистрационных данных) от лиц, не имеющих прав на её получение. **Целостность** — обеспечение уверенности в том, что части системы (программные и данных) не подменены или не модифицированы злоумышленником. **Выполнение обязательств** — подтверждение того, что пользователь действительно совершил определённое действие в системе.

**Аутентификация.** **Последовательность действий.** *Предоставление идентификационных данных*, закреплены за учётной записью и хранятся в рамках подсистемы «Безопасность». К идентификационным данным относят логин и пароль. **Авторизация** (проверка идентификационных данных), результат — либо отклонение, либо предоставление доступа к системе.

**Секретность.** Вся информация должна шифроваться в процессе введения её в систему. Т.е. система хранит важную информацию только в зашифрованном виде. Должна поддерживаться возможность смены пароля к шифру в случае необходимости.

**Целостность.** Блоки хранимой в рамках системы информации должны сопровождаться цифровой подписью.

**Выполнение обязательств.** Фиксирование деятельности пользователя в рамках системы.

## 7. Подсистема «Коммуникации»

**Подсистема «Коммуникации»** — подсистема поддержки *учебного процесса*, отвечающая за совместную работу участников процесса обучения в рамках *занятия*. **Коммуникация** — процесс взаимодействия *участников процесса обучения* в рамках обучающей системы. **Общение** — тип *коммуникации*,

в рамках которой участники обмениваются текстовыми сообщениями (ведут диалог).

**Организация коммуникации.** В пределах *вычислительной машины*. В пределах *локальной сети*. В пределах *глобальной сети*.

## 8. Подсистема «Взаимодействие с системой»

**Подсистема «Взаимодействие с системой»** — подсистема поддержки *учебного процесса*, ответственная за визуализацию процессов, происходящих в процессе общения пользователей с системой. **Графический интерфейс пользователя** — система, наглядно представляющая пользователю функционал обучающей системы и обеспечивающая к нему (функционалу), а также результатам функционирования удобный доступ. **Обложка интерфейса пользователя** — набор сменных элементов оформления *графического интерфейса пользователя*, рисунки, шрифты, размеры, формы. **Конфигурация интерфейса пользователя** — расположение элементов *графического интерфейса пользователя* на экране конкретного пользователя системы.

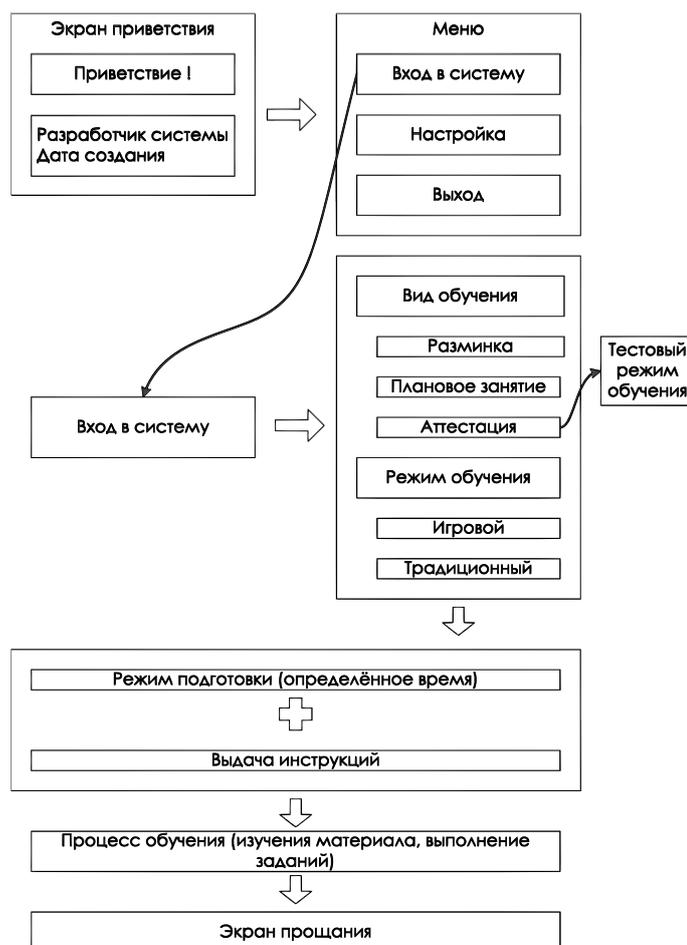


Рис. 7. Взаимодействие с системой

**Экран приветствия.** Включает следующую информацию: *Приветствие пользователя, Информация о создателе.*

**Меню. Вход в обучающую систему.** Выбирается *вид обучения*, варианты: разминка, плановое занятие, аттестация (если выбрана аттестация, сразу осуществляется переход в тестовый режим). Выбирается *режим обучения* на время занятия (во время занятия режим можно сменить). Варианты: игровой, традиционный. Переход в *режим подготовки. Выдача инструкций.* В рамках планового занятия доступен *общий диалог.* Для центрального участника — *добавление участников* в занятие, *распределение по группам, выбор занятия. Настройка системы. Выход из системы.*

## 9. Параметры

Общее количество участников в рамках занятия (одушевлённых):

***LessonParticipantsMaxNum*** (устанавливается наблюдающим-учителем) — максимальное допустимое количество участников в рамках занятия;

***LessonParticipantsMinNum*** (устанавливается наблюдающим-учителем) — минимальное допустимое количество участников в рамках занятия,  $\leq$  ***LessonParticipantsMaxNum***;

***LessonParticipantsEnabledNum*** (рассчитывается системой, для статистики) — общее количество задействованных участников,  $\leq$  ***LessonParticipantsMaxNum*** и  $\geq$  ***LessonParticipantsMinNum***.

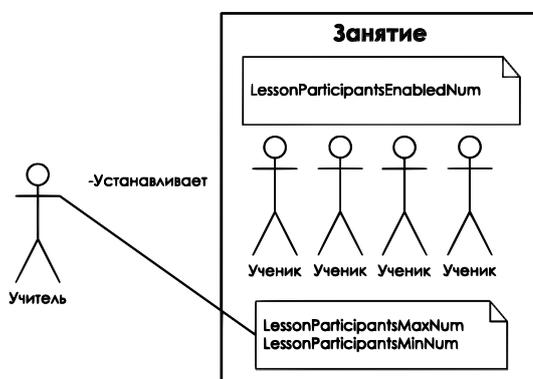


Рис. 8. Занятие

Общее количество участников в рамках экземпляра задачи:

***TaskParticipantsMinNum*** (устанавливается составителем задачи) — минимальное допустимое количество участников в рамках экземпляра задачи;

***TaskParticipantsMaxNum*** (устанавливается составителем задачи) — максимальное допустимое количество участников в рамках экземпляра задачи,  $=$  ***TaskParticipantsMinNum*** в случае, когда задача предполагает определённое количество участников и  $=$  ***TaskParticipantsMinNum*** = 1, когда задача предполагает решение только одним участником, во всех остальных случаях  $\geq$  ***TaskParticipantsMinNum***;

***TaskParticipantsEnabledNum*** (рассчитывается системой, для статистики) — количество задействованных в экземпляре задачи участников  $\leq$  ***TaskParticipantsMaxNum*** и  $\geq$  ***TaskParticipantsMinNum***;  $\leq$   $(PlannedTaskStepsNum / PlannedGroupParticipantTaskSteps)$ , если ***PlannedGroupParticipantTaskSteps***  $\neq$  0 (т.е., когда задача решается в группе);

***TaskOpposedGroupsMaxNum*** (устанавливается наблюдающим-учителем) — максимальное допустимое количество противоборствующих групп в рамках задачи (у каждой группы свой экземпляр одной и той же задачи);

***TaskSpecialPurposeGroupsMaxNum*** (устанавливается наблюдающим-учителем) — максимальное допустимое количество целевых групп в рамках задачи (у каждой группы могут быть разные задачи);

***TaskGroupsEnabledNum*** (рассчитывается системой, для статистики) — количество задействованных групп в рамках задачи  $\leq$  ***TaskOpposedGroupsMaxNum*** и  $\leq$  ***TaskSpecialPurposeGroupsMaxNum***;

***TaskGroupParticipantsMinNum*** (устанавливается наблюдающим-учителем, при условии, что ***TaskGroupParticipantsMinNum***  $\geq$  ***TaskParticipantsMinNum***) — минимальное допустимое количество участников в группе;

***TaskGroupParticipantsMaxNum*** (устанавливается наблюдающим-учителем, при условии, что ***TaskGroupParticipantsMaxNum***  $\leq$  ***TaskParticipantsMaxNum***) — максимальное допустимое количество участников в группе;

***TaskGroupParticipantsEnabledNum*** (рассчитывается системой, для статистики) — количество задействованных участников в рамках конкретной учебной группы  $\geq$  ***TaskGroupParticipantsMinNum*** и  $\leq$  ***TaskGroupParticipantsMaxNum***.

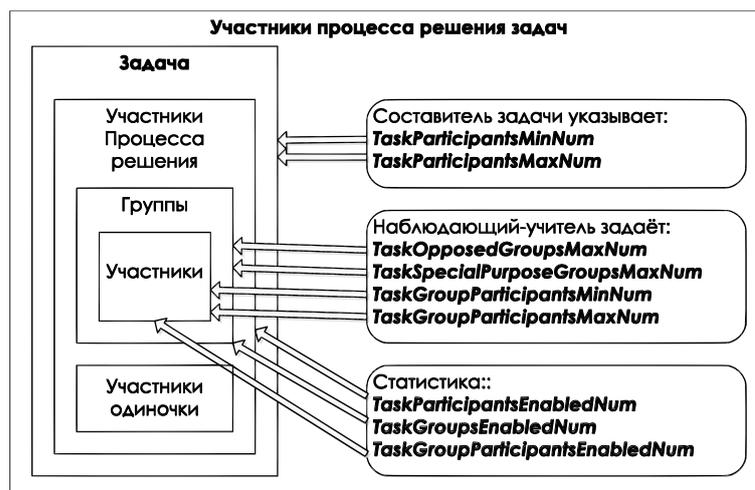


Рис. 9. Участники процесса решения задач

Центральные участники:

***PrincipalParticipantsMaxNum*** (системные ограничения, задаётся администратором) — максимальное допустимое количество;

***PrincipalParticipantsEnabledNum*** (рассчитывается системой, для статистики) — количество задействованных участников  $\leq$  ***PrincipalParticipantsMaxNum***.

Удалённые участники:

***RemoteParticipantsMaxNum*** (системные ограничения, задаётся администратором) — максимальное допустимое количество;

***RemoteParticipantsEnabledNum*** (рассчитывается системой, для статистики) — количество задействованных участников  $\leq$  ***RemoteParticipantsMaxNum***.

Неодушевлённые участники в рамках группы, решающей задачу (реализуемые техническими средствами, добавляются наблюдающим-учителем, если не хватает нужного количества участников; например, согласно условию ***TaskParticipantsMinNum*** или ***TaskGroupParticipantsMinNum***):

***InanimateParticipantsMaxNum*** (устанавливается наблюдающим-учителем) — максимальное допустимое количество  $\leq$  ***TaskGroupParticipantsMaxNum*** — ***TaskGroupParticipantsEnabledNum***, при условии, что ***TaskGroupParticipantsEnabledNum***  $>$  1, т.е. задачу решает как минимум один одушевлённый участник;

***InanimateParticipantsEnabledNum*** (рассчитывается самостоятельно, для статистики) — количество задействованных неодушевлённых участников  $\leq$  ***InanimateParticipantsMaxNum***.

Одушевлённые участники в рамках группы, решающей задачу (обязательно присутствие хотя бы одного одушевлённого участника; в случае участника-одиночки, если задача требует участия определённого количества участников, участнику-одиночке придётся стать членом группы из неодушевлённых участников):

***AnimateParticipantsEnabledNum*** (рассчитывается системой, для статистики) — количество одушевлённых участников,  $\leq$  ***TaskGroupParticipantsMaxNum*** (если участник — член группы), = 1 в случае с участником-одиночкой (когда участник может заниматься самостоятельно за рамками учебного класса).

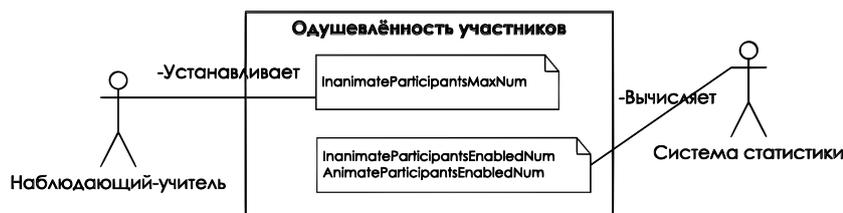


Рис. 10. Одушевлённость участников

Количество разрешённых подсказок (используется один из подходов):  
**HintsLegalNum** (устанавливается составителем задачи) — количество разрешённых подсказок на задачу  $\leq$  **PlannedTaskStepsNum** (если **PlannedTaskStepsNum**  $>$  0);

**HintsLegalNumPartFromPlannedSteps** (устанавливается составителем задачи) — доля подсказок (разрешённых) от количества плановых ходов **PlannedTaskStepsNum**, если **PlannedTaskStepsNum** строго задан.

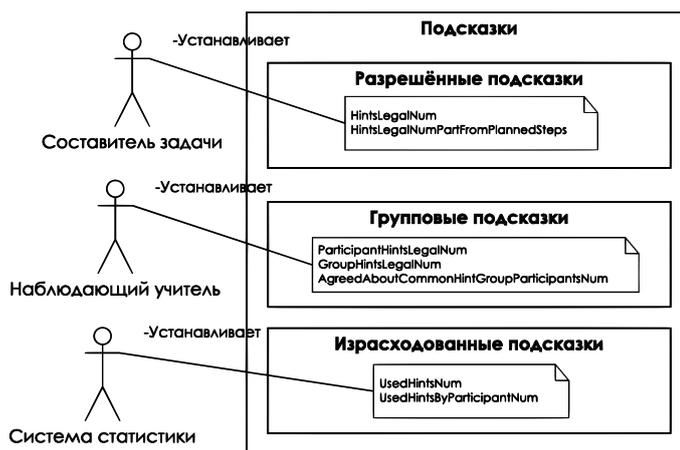


Рис. 11. Подсказки

Подсказки в рамках группы:

**ParticipantHintsLegalNum** (устанавливает наблюдающий-учитель) — количество подсказок на каждого члена группы  $\leq$  **HintsLegalNum**, можно распределять в процессе для конкретных игроков;

**GroupHintsLegalNum** (устанавливает наблюдающий-учитель) — количество подсказок на группу, которыми может воспользоваться любой член группы, при согласии остальных, вместе со всеми **ParticipantHintsLegalNum**  $\leq$  **HintsLegalNum**;

**AgreedAboutCommonHintGroupParticipantsNum** (устанавливается наблюдающим-учителем) — достаточное число членов учебной группы, чьё согласие требуется на разрешение запроса использования групповой подсказки.

Количество израсходованных подсказок:

**UsedHintsNum** (рассчитывается системой, для статистики) — количество израсходованных подсказок в рамках решения задачи;

**UsedHintsByParticipantNum** (рассчитывается системой, для статистики) — количество подсказок, израсходованных участником группы.

Количество разрешённых пропусков (используется один из подходов):

**TaskPassesLegalNum** (устанавливается составителем задачи) — количество разрешённых пропусков на задачу  $\leq$  **PlannedTaskStepsNum**;

***TaskPassesLegalNumFromPlannedSteps*** (устанавливается составителем задачи) — доля пропусков (разрешённых) от количества плановых ходов ***PlannedTaskStepsNum***, если ***PlannedTaskStepsNum*** строго задан.

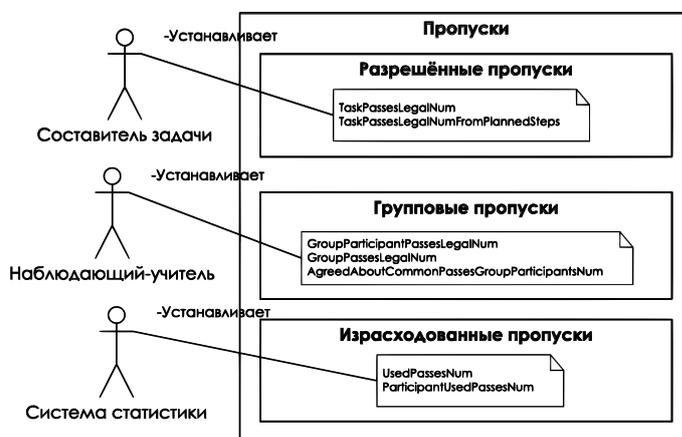


Рис. 12. Пропуски

Пропуски в рамках группы:

***GroupParticipantPassesLegalNum*** (устанавливается наблюдающим-учителем) — количество пропусков на каждого члена группы;

***GroupPassesLegalNum*** (устанавливается наблюдающим-учителем) — количество пропусков на группу, вместе со всеми ***GroupParticipantPassesLegalNum***  $\leq$  ***TaskPassesLegalNum*** (т.е. это пропуски, доступные для всей группы, помимо личных подсказок каждого члена группы, такая же схема и с подсказками);

***AgreedAboutCommonPassesGroupParticipantsNum*** (устанавливается наблюдающим-учителем) — достаточное число членов *учебной группы*, чьё согласие требуется на разрешение запроса использования *группового пропуска*.

Количество израсходованных пропусков:

***UsedPassesNum*** (рассчитывается системой, для статистики) — количество *израсходованных* пропусков;

***ParticipantUsedPassesNum*** (рассчитывается системой, для статистики) — количество пропусков, израсходованных участником.

Общие ходы на задачу:

*Примечание: если нет групп, то используются только эти показатели.*

***PlannedTaskStepsNum*** (устанавливается составителем задачи) — общее количество возможных *плановых* ходов на задачу;

***UsedTaskStepsNum*** (рассчитывается системой, для статистики) — общее

количество *совершенных* в рамках процесса решения задачи ходов.

Групповые ходы на задачу:

*Примечание: если есть группы, то помимо общих показателей применяются ещё и групповые.*

***PlannedGroupParticipantTaskSteps*** (устанавливается составителем задачи) — количество *плановых* ходов на задачу, отведённое *под одного участника*;

***UsedGroupParticipantTaskSteps*** (рассчитывается системой, для статистики) — количество *совершенных конкретным участником* ходов.

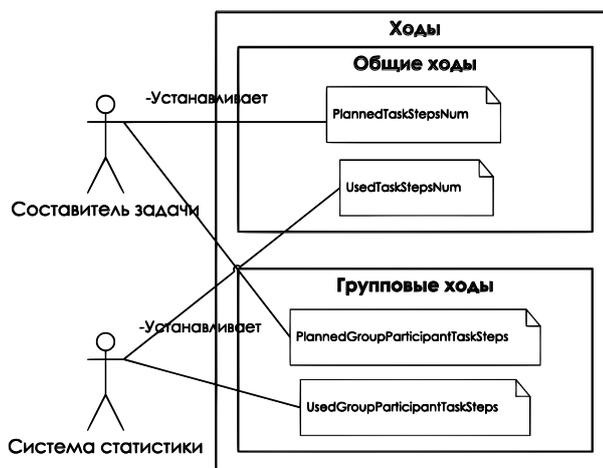


Рис. 13. Ходы

Игровой режим, игровое поле:

***InitialConditionLexemeLength*** (устанавливается составителем задачи) — длина лексемы начального условия в литерах/клетках;

***GameFieldHeight*** (устанавливается составителем задачи) — высота игрового поля в клетках;

***GameFieldWidth*** (устанавливается составителем задачи) — ширина игрового поля в клетках;

***NotUsedGameFieldCellsNum*** (рассчитывается самостоятельно, для статистики) — количество неиспользуемых клеток;

***GameFieldSize*** (рассчитывается самостоятельно, для статистики) — размер игрового поля в клетках  $\leq \mathbf{GameFieldHeight} * \mathbf{GameFieldWidth} - \mathbf{NotUsedGameFieldCellsNum}$ ;

***StepLexemeLength*** (рассчитывается самостоятельно, для статистики) — длина лексемы, составленной за один ход.

Традиционный и тестовый режимы:

***QuestionChoiseTaskTestUnitNum*** (устанавливается составителем задачи) — количество проверочных единиц в задаче типа *вопрос — выбор ответа*;

***QuestionWriteTaskTestUnitNum*** (устанавливается составителем задачи) —

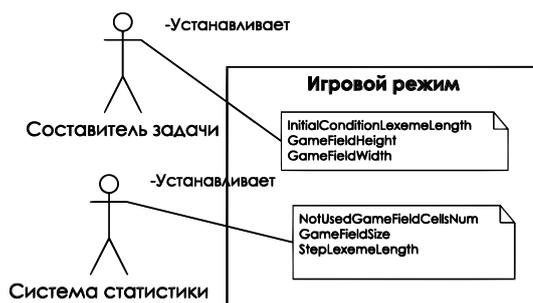


Рис. 14. Игровой режим

количество проверочных единиц в задаче типа *вопрос — запись ответа*.

Тестовый режим:

***QuestionSpecialWriteTaskTestUnitNum*** (устанавливается составителем задачи) — количество проверочных единиц в задаче типа *вопрос — запись развернутого ответа*.

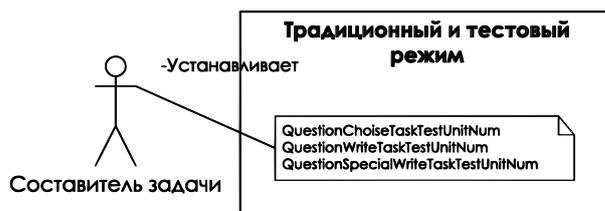


Рис. 15. Традиционный и тестовый режим

Очки:

***StepUnitScores*** (устанавливается составителем задачи) — бал единицы хода (количество очков, начисляемых за составляющую хода или за весь ход).

*Пример: стоимость литеры в составленной лексеме, или стоимость составленной лексемы.*

Суммы:

***ParticipantStepsScoresSum*** (рассчитывается самостоятельно, для статистики) — сумма очков, полученных за все игровые ходы *конкретным игроком*;

***GroupStepsScoresSum*** (рассчитывается самостоятельно, для статистики) — сумма очков, полученных за все игровые ходы *группой игроков*.

Время задачи:

***TaskPlannedTime*** (устанавливается составителем задачи) — *предполагаемое* время;

***TaskMaxTime*** (устанавливается наблюдающим-учителем) — *максимально допустимое* время, устанавливается в ориентации на ***TaskPlannedTime***;

***TaskUsedTime*** (рассчитывается системой, для статистики) — фактическое, *затраченное* время.

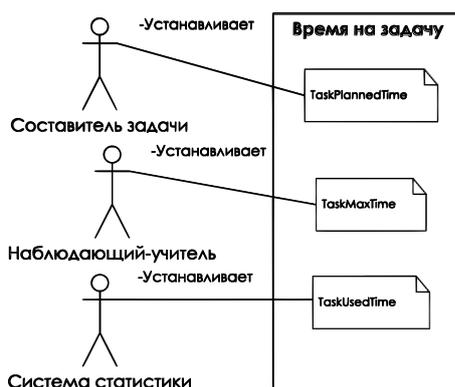


Рис. 16. Время на задачу

## 10. Виды обучения

**Разминка.** Выбор занятия по теме, выбор режима (игрового или традиционного).

*Примечание:* занятие может быть как проверочным, так и учебным.

*Примечание:* организационная особенность участника — центральный участник.

$TaskParticipantsMaxNum = 1$ .

**Плановое обучение.** Занятия вместе с учебным классом.

*Примечание:* организационная особенность — удалённый участник или центральный участник (может отсутствовать наблюдающий-учитель).

**Аттестация.** Выбор экзамена.

*Примечание:* организационная роль участника — удалённый участник. В роли центрального участника — наблюдающий-учитель.

*Примечание:* режим обучения — тестовый.

## 11. Режимы обучения

Общие особенности неигровых режимов:

В контексте неигрового режима шаг называется **проверочной единицей**. Существуют следующие виды. **Вопрос — варианты ответов.** Среди предложенных вариантов может быть только один правильный, либо несколько правильных. **Вопрос — ответ от ученика.** Ответ может быть *сжатым*, представленным в виде понятия или словосочетания, либо же *развёрнутым*. Последний вид ответа оценивается без участия технических средств. Характеристика любого вида ответа — его стоимость.

Результат представляется в виде сравнения правильных вариантов хода (если правильных вариантов несколько, каждому из вариантов присваивается определённый вес) и совершённых участником вариантов хода. Оценка пред-

Выбор вида обучения	
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Разминка</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center; margin: 0;"><b>Доступное занятие</b> (тема, дата)</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Учебное</div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Проверочное</div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Режим</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Игровой</div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Традиционный</div> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;"><b>Аттестация</b></p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="margin: 0;">Режим обучения: тестовый; Орг. особенность: удалённый участник; Активна, если открыта наблюдающим-учителем</p> </div> </div>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Плановое обучение</b></p> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 5px 0;"> <p style="margin: 0;">Орг. особенность: удалённый участник или центральный;</p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center; margin: 0;"><b>Актуальное занятие</b></p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Повторение изученного</div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Учебное</div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Проверочное</div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; margin: 5px 0;"> <p style="margin: 0;">Соответствующий пункт деактивируется по усмотрению наблюдающего-учителя (если он есть)</p> </div> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center; margin: 0;"><b>Режим</b></p> </div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Игровой</div> <div style="border: 1px solid black; padding: 5px; margin: 5px 0; text-align: center;">Традиционный</div>

Рис. 17. Выбор вида обучения

ставляется в виде отношение веса полностью правильных ходов к весу осуществлённых ходов.

Статистика решения задачи. **Ходы.** Представлены на показ *совершённые* ходы и *корректные* (каким ход должен быть в идеале), *затраченное время*, *весовые характеристики* как корректного хода, так и совершённого. **Результат.** Производится выбор *балльной системы*. Варианты: традиционная пятибалльная (2..5), десятичная (0..10), стобалльная (0..100), последняя используется по умолчанию.

Особенности игрового режима:

Примечание: игровой режим возможен в рамках учебного и проверочного занятия.

Примечание: занятие идёт по схеме повторяющейся последовательности (изучение фрагмента учебного материала, игра).

**Подсказки и пропуски хода.** По умолчанию полностью разрешены. **Оценивание результата.** Объектом оценки выступает *ход*. Формула оценочная, должна учитывать коэффициент **StepUnitScores**. Объектом оценки также выступает сама *игра*. В формуле вычисления оценки для конкретного игрока используются значения: **ParticipantStepsScoresSum**. Для учебной группы иг-

роков используются значения: **GroupStepsScoresSum**. Конечное решение в оценивании результата остаётся за *наблюдающим-учителем* или *судьёй*.

**Игровое поле.** Имеет прямоугольную, клеточную форму.

Особенности традиционного режима:

*Примечание:* режим возможен в рамках учебного занятия.

*Примечание:* занятие идёт по схеме повторяющейся последовательности (изучение фрагмента учебного материала, ответы на заданные вопросы).

**Задания** бывают следующих видов, *вопрос — выбор ответа* и *вопрос — запись ответа*. **Деление** задач (не заданий) осуществляется по теме и сложности.

Особенности тестового режима:

*Примечание:* тестовый режим возможен в рамках проверочного занятия или аттестации.

*Примечание:* занятие идёт по схеме повторяющейся последовательности (*вопрос, ответ*).

**Задачи** могут делиться по признаку, например, тематическому. В рамках задач даются связанные задания. **Типы заданий.** *Вопрос — выбор ответа*. *Вопрос — запись ответа*. *Вопрос — запись развёрнутого ответа*, оценка которого производится без участия технических средств.

## 12. Диалоги

*Примечание:* диалог представлен в виде обмена текстовыми сообщениями (*чатом*).

**Чат.** *Общий чат.* Организация письменного общения между всеми участниками процесса обучения. *Групповой чат.* Организация общения между всеми участниками учебной группы. *Приватный чат.* Организация общения между конкретными участниками процесса обучения.

*Примечание:* чат возможен только между одушевлёнными участниками.

## Заключение

В статье представлена концепция проекта расширяемой обучающей программной системы. Выделены возможные элементы и подсистемы. Предложены параметры настройки и конфигурирования ключевых процессов.

## ЛИТЕРАТУРА

1. Гусс С.В. Элементы повторного использования для программных систем учебного назначения. Концептуальное проектирование и анализ // Математические структуры и моделирование. 2009. Вып. 20. С. 8–92.

2. Гусс С.В. Проблема повторного использования в разработке семейства игровых программных систем учебного назначения // Вестник Омского университета. 2010. № 4. С. 147–149.
3. Гусс С.В. Предметно-ориентированные проектные решения для тематической области обучающих программных средств на основе лингвистических игр // Математические структуры и моделирование. 2011. Вып. 24. С. 55–68.

## **УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ВЫЧИСЛИТЕЛЬНЫЙ МОДУЛЬ С АРХИТЕКТУРОЙ ПРОЦЕССОРА ALPHA**

**С.В. Стопкин, К.В. Пугин, Д.М. Бречка**

Данная работа посвящена выбору и установке программного обеспечения на вычислительный модуль с архитектурой процессора Alpha. Вычислительные модули с такими процессорами входят в состав кластера МВС-1000/128, находящегося в распоряжении факультета компьютерных наук ФГБОУ ВПО «ОмГУ им. Ф.М. Достоевского». В результате работы на вычислительный модуль была установлена операционная система Debian Linux, а также средство поддержки исполнения параллельных программ — OpenMPI.

### **Введение**

Вычислительный кластер МВС–1000/128 поступил в распоряжение факультета компьютерных наук ФГБОУ ВПО «ОмГУ им. Ф.М. Достоевского» в июле 2011 года. Кластер состоит из 64 вычислительных модулей, связанных между собой сетью на основе 100baseT Ethernet. Каждый вычислительный модуль содержит в себе два процессора DEC Alpha 21164 с частотой 667 МГц. Общий объем оперативной памяти модуля составляет 2048 Мбайт.

На каждом вычислительном модуле установлена операционная система RedHat Linux Alpha 7.2 и библиотека функций передачи сообщений MPICH. Также для пользователя доступны компиляторы Compaq Fortran, Compaq C/C++ и библиотека Compaq Extended Math Library (CXML). Для организации пакетной обработки применяется система управления прохождением заданий, разработанная в ИПМ им. Келдыша РАН.

Программное обеспечение, установленное на модулях кластера, на сегодняшний день можно считать устаревшим. Для организации эффективной работы предлагается установить на вычислительные модули более современные программные продукты.

Целью данной работы является выбор подходящей операционной системы и

ее установка на вычислительный модуль кластера, а также установка современных компиляторов и средств поддержки исполнения параллельных программ.

## 1. Описание аппаратной платформы и проблема выбора программного обеспечения

Подробное описание аппаратного обеспечения содержится в таблице 1 [1,2]. Основная проблема выбора программного обеспечения заключается в том, что архитектура Alpha не поддерживается большинством современных операционных систем, так как считается устаревшей (процессоры Alpha не выпускаются с 2004 года). Главным вопросом выбора программного обеспечения является выбор операционной системы.

Таблица 1. Описание аппаратного обеспечения вычислительного модуля

Тип микропроцессора	DEC Alpha 21164
Разрядность микропроцессора	64
Набор инструкций	RISC
Количество ядер микропроцессора	1
Частота микропроцессора	667 МГц
Напряжение питания микропроцессора	2 В
Объем кэша данных	64 Кбайт
Объем кэша команд	64 Кбайт
Количество микропроцессоров на плате	2
Количество оперативной памяти	2048 Мбайт
Тип оперативной памяти	DIMM SDRAM PC100+ ECC
Количество памяти, доступной микропроцессору	2048 Мбайт
Тип материнской платы	Api UP2000
Наличие коммуникационных портов	LPT, 2 x COM, SCSI, IDE, 2 x PS/2
Тип сетевого адаптера	PCI - 100BaseT
Тип графического адаптера	PCI VGA 4 Мбайт
Тип жесткого диска	IDE ATA133
Объем жесткого диска	20 Гбайт
Максимальная потребляемая мощность	600 Вт

## 2. Выбор операционной системы

В ходе работы был выявлен ряд операционных систем, которые поддерживают архитектуру Alpha. Некоторые из этих систем продолжают поддержку

Alpha и на сегодняшний день, в других — поддержка завершена. Классификация систем приведена на рисунке [3–5].

При выборе операционной системы для установки на вычислительный модуль рассматривались только те системы, поддержка архитектуры Alpha для которых продолжается и сегодня. Кроме того, рассматривались только свободные операционные системы. Выбор остановился на системе Debian Linux, основной причиной выбора именно этой системы является её хорошая документированность.

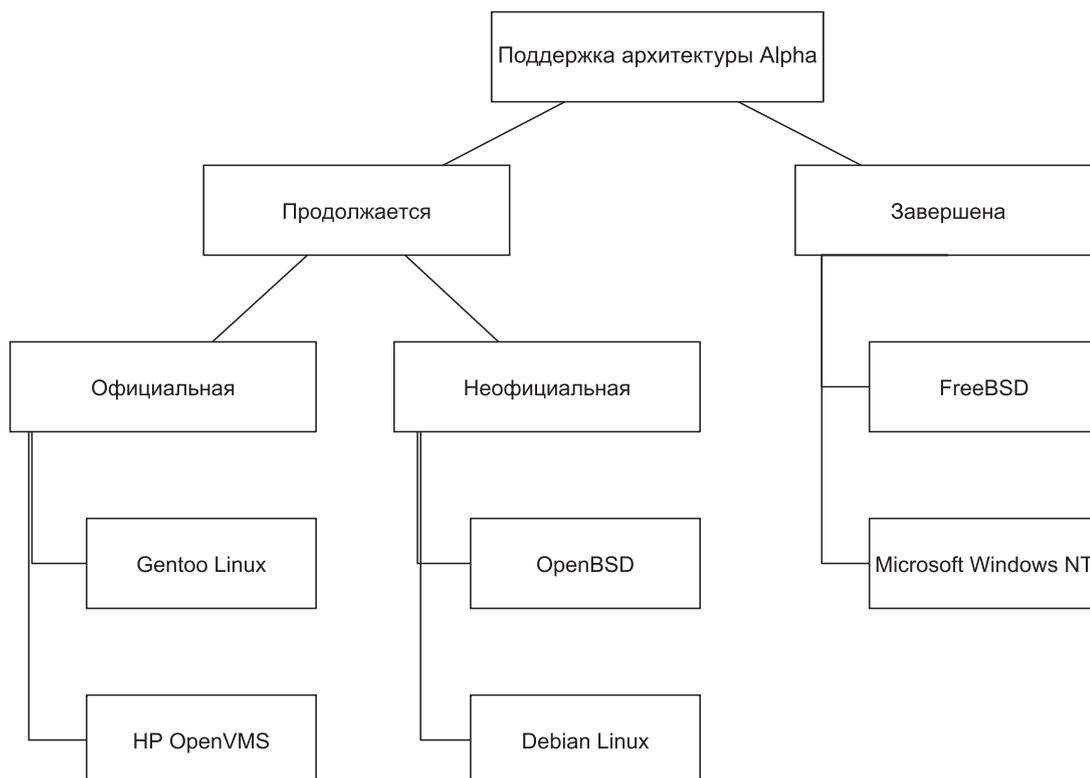


Рис. 1. Операционные системы, поддерживающие архитектуру Alpha

### 3. Установка Debian Linux на вычислительный модуль

#### 3.1. Встроенное программное обеспечение

Для компьютерных систем на базе микропроцессоров Alpha доступны два класса встроенного программного обеспечения [2, 4]:

1. SRM-консоль, основанная на спецификации Alpha Console Subsystem, которая обеспечивает рабочую среду для операционных систем OpenVMS, Tru64 UNIX и Linux;
2. ARC-консоль, AlphaBIOS или ARCSBIOS, основанная на спецификации Advanced RISC Computing (ARC), которая обеспечивает рабочую среду для Windows NT.

С точки зрения пользователя наиболее важное различие между SRM и ARC — это различные схемы разметки жёсткого диска, с которого планируется загрузка, и интерфейс системы ввода-вывода. В вычислительных модулях на базе материнской платы Арі UP2000 в качестве встроенного ПО используется консоль SRM [5–9]. Опишем основные особенности этой программы.

SRM-консоль — это микропрограмма (аналогичная BIOS) для компьютерных систем с процессорами Alpha. Интерфейс SRM выполнен в стиле Unix shell. Микропрограмма хранится во флэш-ПЗУ и выполняется при включении и/или перезагрузке системы Alpha.

SRM-консоль поддерживает загрузку с FDD, HDD, CD/DVD и сети. На режим сетевой загрузки накладываются аппаратные и программные ограничения. Поскольку Alpha не поддерживает для HDD таблицу разделов MS-DOS, для загрузки необходим HDD с таблицей разделов в формате BSD. Консоль сама не в состоянии загрузить Linux, для этого требуется промежуточный системный загрузчик. Для SRM консоли таким загрузчиком является aboot — многоплатформенный системный загрузчик [9].

Загрузчик aboot занимается трансляцией команд SRM в команды, передаваемые ядру Linux, а также загрузкой ядра и начального образа в ОЗУ. Конфигурация aboot содержится в файле /etc/abootconf, в этом же файле хранится список возможных конфигураций загрузки ядра. Конфигурация ядра передаётся в загрузчик при помощи параметра -flags в SRM. Программа aboot имеет собственную командную строку, которая позволяет менять на ходу конфигурацию загрузки ядра. Данная возможность может быть полезна при проверке корректности обновлений ядра.

### 3.2. Выбор версии Debian Linux

Релизы Debian Linux разделены на пять веток [5]:

1. Ветка oldstable — содержит пакеты предыдущего стабильного дистрибутива.
2. Ветка stable (стабильная) — содержит пакеты, вошедшие в последний официальный дистрибутив (обновление пакетов в нём происходит только для устранения уязвимостей).
3. Ветка testing (тестируемая) — из этой ветки будет формироваться следующий стабильный дистрибутив.
4. Ветка unstable (нестабильная) — пакеты этой ветки готовятся к помещению в тестируемую ветку.
5. Ветка experimental (экспериментальная) — не является полноценной веткой, в ней находятся пакеты, требующие тщательного тестирования или пакеты, которые повлекут серьёзные изменения в дистрибутиве.

На момент проведения исследовательских работ, последним релизом Debian Linux была версия 6.0 с официальным названием Squeeze. Однако, поддержка архитектуры Alpha сообществом Debian была прекращена в версии Debian 5.0 Lenny. Поэтому именно Debian 5.0 Lenny будет использоваться для установки на вычислительный модуль кластера.

### **3.3. Установка базовой конфигурации Debian 5.0 Lenny и дополнительных программных пакетов**

Наиболее простым способом установки операционной системы с помощью SRM является установка системы с DVD-носителя. Именно этот способ установки и был использован. Опишем процесс установки. После помещения DVD с дистрибутивом системы в дисковод необходимо нажать Ctrl+P для вызова SRM-консоли. В консоли при помощи команды `show dev` (показывает список всех дисковых устройств на модуле) можно узнать номер дисковода, этот номер передаётся в качестве параметра команде `boot` для начала загрузки системы с DVD. Далее установка проходит в штатном режиме, подробное описание установки Debian Linux приводится в [5].

После установки базовой конфигурации системы для более удобной организации работы с ней были добавлены следующие программные пакеты:

1. Пакет `mc` — midnight commander, консольный двухпанельный файловый менеджер с поддержкой `ftp`, `scp`, `ssh`.
2. Пакет `ssh` — `openssh`, набор клиент-серверных утилит для защищённого соединения с удалённым компьютером.
3. Пакет `htop` — монитор процессов для Linux с отображением загрузки CPU и потребления памяти.
4. Пакет `tmux` — свободная консольная утилита-мультиплексор, предоставляющая пользователю доступ к нескольким терминалам в рамках одного экрана.
5. Пакет `GCC` — коллекции компиляторов GNU.

## **4. Обновление Debian 5.0 Lenny и установка средств поддержки параллельных программ**

Для организации параллельных вычислений на кластере требуется система распределённых вычислений MPI. Было принято решение использовать реализацию OpenMPI, так как разработка более старой реализации — MPICH — на сегодняшний день прекращена [10]. Однако при установке OpenMPI на систему Debian 5.0 Lenny был выявлен ряд проблем, связанных с тем, что Lenny использует относительно устаревшие версии некоторых программных пакетов. В связи с этим было принято решение обновить операционную систему до более актуальной версии.

Попытка обновить систему до версии Squeeze по руководствам сообщества Debian [5] привела к невозможности загрузки операционной системы. Эта ситуация, по-видимому, связана с наличием некоторых ошибок в пакетном менеджере `dpkg`, используемом в Debian Linux. Основная проблема была связана с обновлением архиватора `tar`, использующегося для архивации установочных пакетов, с которыми работает пакетный менеджер. После ручной замены исполняемых файлов `tar` на неконфликтующие и обновления пакетного менеджера `dpkg` до версии, не имеющей ошибки, была получена возможность корректного автоматического обновления системы до версии Squeeze. При этом ядро системы обновить не удалось, так как в новом ядре отсутствует поддержка платформы Alpha.

После успешно проведенного обновления системы до версии Squeeze было решено осуществить попытку совершить ещё одно обновление системы до версии Sid для получения самых современных версий программных пакетов. Надо отметить, что на момент проведения работ дистрибутив Debian Sid находился в ветке `unstable`.

При обновлении до Debian Sid происходит замена многих системных средств на новые, что привело к следующим проблемам:

1. Ядро системы версии 2.6.32 по умолчанию ссылкам на устройства дисков даёт имена типа `sdaX`, но для работы загрузчика `aboot` необходимо, чтобы имена имели вид `hdaX`. Данная проблема решена путём сохранения старого ядра. При установке пакетов в конфигураторе есть возможность отказаться от установки нового ядра.
2. Debian Sid по умолчанию использует загрузку по зависимостям в отличие от традиционной последовательной загрузки, что приводит к проблемам инициализации видеокарты. Проблема решена путём сохранения последовательной загрузки (при помощи конфигуратора).
3. Системная программа динамической генерации фалов устройств Udev в версиях выше 175 не может определить устройства ввода данных модулей. Проблема решена отказом от обновления Udev в составе начального образа системы.
4. Сервер графического окружения X.org не способен определить устройства ввода. Данная проблема находится в стадии решения.

Установка `OpenMPI` как в Debian Squeeze, так и в Debian Sid не вызывает особых затруднений, требуется просто установить пакеты `openmpi` и `openssh` с помощью стандартного пакетного менеджера.

Работоспособность полученной конфигурации была проверена путём компиляции и запуска ряда простых параллельных программ.

## Заключение

В данной статье была рассмотрена задача выбора и установки программного обеспечения на вычислительный модуль кластерной системы МВС-1000/128. Основная проблема при решении данной задачи заключалась в том, что вычислительный модуль имеет весьма специфическую и во многом устаревшую аппаратную платформу. В ходе работы был проведён выбор подходящей операционной системы и программного обеспечения для поддержки исполнения параллельных программ. Выбранная операционная система и необходимые программные пакеты были успешно установлены на вычислительный модуль. Работоспособность модуля была протестирована.

Основным результатом работы является получение работоспособного вычислительного модуля, поддерживающего современную среду исполнения параллельных программ — OpenMPI. Дальнейшие работы будут направлены на создание кластера, состоящего из подобных модулей и тестирование полученного кластера.

## ЛИТЕРАТУРА

1. Archived technical documentation library:[сайт]. URL: <http://h18002.www1.hp.com/alphaserver/technology/chip-docs.html> (дата обращения: 15.10.2011).
2. P. V. Bolotoff Alpha: The History in Facts and Comments: [Электронный документ]. URL: [http://alasir.com/articles/alpha\\_history/alpha\\_21064\\_21066.html](http://alasir.com/articles/alpha_history/alpha_21064_21066.html) (дата обращения: 16.10.2011).
3. Tru64.org: [сайт]. URL: <http://www.tru64.org/> (дата обращения: 14.11.2011).
4. OpenVMS.org: [сайт]. URL: <http://www.openvms.org> (дата обращения: 16.11.2011).
5. Debian: [сайт]. URL: <http://www.debian.org> (дата обращения: 27.11.2011).
6. Red Hat Linux 7.0: The Official Red Hat Linux Alpha Installation Guide: [Электронный документ]. URL: <http://web.archive.org/web/20050408134334/http://www.redhat.com/docs/manuals/linux/RHL-7-Manual/alpha-install-guide/s1-alpha-srm.html> (дата обращения: 11.12.2011).
7. SRM Console Reference: [Электронный документ]. URL: [http://h18002.www1.hp.com/alphaserver/download/srm\\_reference.pdf](http://h18002.www1.hp.com/alphaserver/download/srm_reference.pdf) (дата обращения: 15.12.2011).
8. Alpha SRM Console for Alpha Microprocessor Motherboards. User's Guide: [Электронный документ]. URL: <http://h18002.www1.hp.com/alphaserver/technology/literature/srmcons.pdf> (дата обращения: 15.12.2011).
9. R. Payne, D. Huggins-Daines SRM Firmware Howto: [Электронный документ]. URL: <http://www.tldp.org/HOWTO/SRM-HOWTO/> (дата обращения: 05.01.2012).
10. MPICH 2: [сайт]. URL: <http://www.mcs.anl.gov/research/projects/mpich2/> (дата обращения: 15.03.2012).

## АВТОМАТИЗИРОВАННОЕ ТЕСТИРОВАНИЕ СЛОЖНОСТИ АЛГОРИТМОВ С ПОМОЩЬЮ МОСК-ОБЪЕКТОВ

**Е.А. Тюменцев**

В статье описывается способ тестирования алгоритмической сложности алгоритмов на основе Моск-объектов, позволяющий автоматизировать проверку, исследовать и классифицировать алгоритмы с точки зрения сложности, даже если их реализация неизвестна.

Практически для любой задачи существуют различные способы решения, которые отличаются друг от друга алгоритмической сложностью. Например, в классе двунаправленный список  $\text{list}\langle T \rangle$  элементов типа  $T$  есть метод  $\text{int Count}()$ , который возвращает количество элементов в этом списке. Существуют две очевидные возможности (хотя вариантов, на самом деле, больше):

- Класс  $\text{list}\langle T \rangle$  хранит количество элементов списка в специально отведённом для этого поле. Тогда метод  $\text{int Count}()$  будет иметь сложность  $O(C)$ , где  $C$  — константа.
- Метод  $\text{int Count}()$  перебирает все элементы списка для подсчёта количества элементов. Тогда это метод будет иметь сложность  $O(n)$ , где  $n$  — длина списка.

Чтобы выбирать правильные способы решения, программисту необходимо знать их алгоритмическую сложность. Наиболее распространённым методом оценки сложности является её явное вычисление по исходному коду. К сожалению, исходный код доступен не всегда, а такой метод требует определённой квалификации от программиста и ручной работы. В настоящей статье будет описан подход к написанию автоматизированных тестов на алгоритмическую сложность программного кода на основе Моск-объектов.

**Определение 1** (Моск-объект). Объект-имитация реального объекта программного окружения, реализующая только некоторые аспекты реального объекта с целью тестирования определённого поведения программного окружения.

Mock-объекты не являются библиотекой или конкретной технологией тестирования, они представляют собой сложившуюся методологию тестирования объектно-ориентированного программного обеспечения, основанную на принципе обращения зависимостей (The Dependency Inversion Principle) [1]. Подробнее об истории выработки практики применения Mock-объектов можно прочитать в [2].

Суть предлагаемого способа тестирования можно выразить следующей цепочкой утверждений:

- Алгоритмическая сложность прямо зависит от количества и состава операций, выполненных над входным набором данных.
- Операция в объектно-ориентированном программировании представляет собой поведение некоторого объекта.
- Поведение объекта описывает какой-либо интерфейс.
- Mock-объект подменяет собой реальный объект с целью проведения тестирования.
- Подмена реального объекта Mock-объектом происходит за счёт того, что оба этих объекта реализуют один и тот же интерфейс, а программа в целом должна удовлетворять принципу подстановки Лисков [3].
- Следовательно, чтобы определить алгоритмическую сложность какого-либо алгоритма, необходимо предоставить этому алгоритму вместо реальных объектов Mock-объекты. Каждый метод Mock-объекта будет считать каждый собственный вызов. После того, как тестируемый алгоритм завершит свою работу, будет известно, сколько каждый метод был вызван.
- Остаётся только проверить значения соответствующих счётчиков в зависимости от количества входных данных.

Проиллюстрируем вычисление алгоритмической сложности на примерах. Для примеров был использован C++ и компилятор MSVS 2010. Предположим, что у нас есть бинарное дерево с операцией вставки Insert.

```
template<typename T> class Tree
{
    struct TreeNode
    {
        TreeNode(T const& value)
        {
            this->value = value;
            left = 0;
            right = 0;
        }
        TreeNode* left;
```

```
    TreeNode* right;
    T value;
};

public:

    Tree() { root = 0; }
    void Insert(T const& value)
    {
        if(0 == root)
            root = new TreeNode(value);
        else
            _insert(root, value);
    }

private:

    void _insert(TreeNode *root, T const & value)
    {
        if(value == root->value)
            return;
        else
        {
            if(value < root->value)
            {
                if(0 == root->left)
                    root->left = new TreeNode(value);
                else
                    _insert(root->left, value);
            }
            else
            {
                if(0 == root->right)
                    root->right = new TreeNode(value);
                else
                    _insert(root->right, value);
            }
        }
    }

    TreeNode *root;
};
```

Мы хотим протестировать сложность вставки в данное дерево. Для этого создадим Моск-объект — класс А. Сложность будем считать в операциях чте-

ния и записи данного объекта (при необходимости можно было бы считать и каждую операцию в отдельности). К пишущим операциям отнесём конструктор копии и оператор присваивания. К читающим операциям - ==, < .

```
class A
{
public:
    A(): someValue(0){}
    A(int value): someValue(value) {}
    A(A const& other)
    {
        ++other.writeCounter;
        this->someValue = other.someValue;
    }
    A& operator=(A const& other)
    {
        ++other.writeCounter;
        this->someValue = other.someValue;
        return *this;
    }
private:
    int someValue;
public:
    static int readCounter;
    static int writeCounter;

    static void Clear()
    {
        readCounter = 0;
        writeCounter = 0;
    }

    friend bool operator==(A const& a1, A const & a2);
    friend bool operator<(A const& a1, A const & a2);
};

int A::readCounter = 0;
int A::writeCounter = 0;

bool operator==(A const& a1, A const & a2)
{
    ++a1.readCounter;
    return a1.someValue == a2.someValue;
}
```

```
bool operator<(A const& a1, A const & a2)
{
    ++a1.readCounter;
    return a1.someValue < a2.someValue;
}
```

```
bool operator==(std::pair<const A, int> & p1,
                std::pair<const A, int> & p2)
{
    return p1.first == p2.first;
}
```

Хорошо известно, что в худшем случае вставка  $n$  элементов в бинарное дерево поиска имеет сложность порядка  $O(n^2)$ . Проверим это на примере:

```
Tree<A> tree;
for(int i = 0; i < 1024; ++i)
{
    tree.Insert(A(i));
}
```

```
cout << "Insert 1024 items to nonbalanced tree: read "
      << A::readCounter << " write "
      << A::writeCounter << endl;
```

Вывод на консоль:

```
Insert 1024 items to nonbalanced tree:
read 1047552 write 1024
```

Получаем  $n * (n - 1)$  операций чтения, и  $n$  операций записи, что в сумме даёт  $n^2$  общего числа операций.

В наилучшем случае вставка  $n$  элементов в бинарное дерево поиска имеет сложность порядка  $O(n * \log_2(n))$ . Напишем специальную функцию вставки, которая нам обеспечит наилучший случай:

```
void Insert(Tree<A> & t, int l, int r)
{
    int val = (l+r)/2;
    t.Insert(A(val));

    if(l!= val)
        Insert(t, l, val);
    if(r - val > 1)
        Insert(t, val, r);
    else
        if(r-val == 1)
            t.Insert(r);
}
```

Теперь проверим количество операций:

```
Tree<A> tree;
Insert(tree, 0, 1023);
cout << "Insert 1024 items to nonbalanced tree: read "
      << A::readCounter << " write "
      << A::writeCounter << endl;
```

Вывод на консоль:

```
Insert 1024 items to nonbalanced tree:
read 41483 write 1024
```

$\log_2(1024) = 10$ , а, следовательно, исследуемый нами случай имеет сложность порядка  $C * O(n * \log_2(n))$ .

Очевидно, что рассмотренный код несложно преобразовать в автоматический тест на базе одной из известных библиотек для организации тестирования.

Таким способом можно тестировать и исследовать не только собственные классы, но и сторонний код. Рассмотрим классы `list<T>` и `map<Key, Value>` стандартной библиотеки STL языка C++. Заметим, что в качестве Mock-объекта можно использовать тот же самый класс `A`, который применялся для тестирования класса `Tree`.

Известно, что вставка в список имеет порядок  $O(C)$ .

```
list<A> list;
for(int i = 999; i >= 0; --i)
  list.insert(list.begin(), A(i));
cout << "Insert to list: read " << A::readCounter
      << " write " << A::writeCounter << endl;
```

Вывод на консоль:

```
Insert to list: read 0 write 1000.
```

В худшем случае операция поиска в линейном списке имеет порядок  $O(n)$ .

```
find(list.begin(), list.end(), A(999));
cout << "Find in list: read " << A::readCounter
      << " write " << A::writeCounter << endl;
```

Вывод на консоль:

```
Find in list: read 1000 write 0.
```

Вставка  $n$  элементов в `map` имеет сложность порядка  $O(n * \log_2(n))$ .

```
map<A,int> map;
for(int i = 0; i < 1024; ++i)
  map[A(i)] = i;
cout << "Insert to map 1024 items: read " << A::readCounter
      << " write " << A::writeCounter << endl;
```

Вывод на консоль:

```
Insert to map 1024 items: read 31828 write 2048
```

По выводу на консоль видно, что при вставке в `map` вставляется не сам элемент, а его копия, что для больших объектов без разделения состояния между экземплярами может привести к накладным расходам, сводящим на нет преимущества от использования `map`.

В настоящей статье был рассмотрен подход к реализации тестирования алгоритмической сложности объектно-ориентированных систем с применением `Mock`-объектов, позволяющий:

- автоматизировать проверку алгоритмической сложности используемого решения;
- вычислять трудоёмкость в терминах выполняемых операций;
- как следствие предыдущего пункта, применять количественные критерии, не зависящие от вычислительного устройства, при описании требований к производительности;
- исследовать и классифицировать алгоритмы с точки зрения сложности, даже если их реализация неизвестна;
- применять разработанные тесты и на реальных данных.

## ЛИТЕРАТУРА

1. The Dependency Inversion Principle. URL: <http://www.objectmentor.com/resources/articles/dip.pdf> (дата обращения: 01.12.12)
2. A Brief History of Mock Objects. URL: <http://www.mockobjects.com/2009/09/brief-history-of-mock-objects.html> (дата обращения: 01.12.12)
3. The Liskov Substitution Principle. URL: <http://www.objectmentor.com/resources/articles/lsp.pdf> (дата обращения: 01.12.12)

## **ОБЗОР РЕШЕНИЙ ДЛЯ РЕАЛИЗАЦИИ ПРОТОТИПА ЗАМКА МОБИЛЬНОЙ БИОМЕТРИЧЕСКОЙ СИСТЕМЫ ДОСТУПА M-BLOCK**

**А.А. Коровянский**

В статье рассматриваются существующие средства для реализации прототипа замка мобильной биометрической системы доступа M-Block. В результате сравнительного анализа определяется наиболее подходящее решение.

### **Введение**

При реализации любого программно-аппаратного комплекса большое значение имеет обзор, сравнительный анализ и выбор наиболее подходящего средства для реализации. Мотивацией написания данной статьи является решение данной задачи для прототипа замка мобильной биометрической системы доступа M-Block, разрабатываемой на факультете компьютерных наук ОмГУ им. Ф.М. Достоевского.

Биометрическая система доступа (БСД) — это программно-аппаратный комплекс, решающий задачу разграничения физического доступа к целевому объекту путём проверки биометрических данных. Таким образом, для открытия дверей используются специальные замки и биометрические алгоритмы, верифицирующие человека по отпечатку пальца, внешности, голосу или другому биометрическому фактору.

В основе M-Block лежит идея использования возможностей смартфонов и других современных мобильных устройств для создания инновационной и промышленно значимой мобильной БСД. Система должна обеспечивать высокий уровень безопасности, поддерживать различные факторы биометрической авторизации и иметь возможность централизованной и простой настройки политик доступа.

На рис 1. представлена принципиальная схема M-Block. Три ключевых компонента системы: мобильное приложение, приложение замка и приложение сервера.

Последовательность событий при взаимодействии пользователя с системой:

1. Пользователь подходит к двери с целью открыть её.

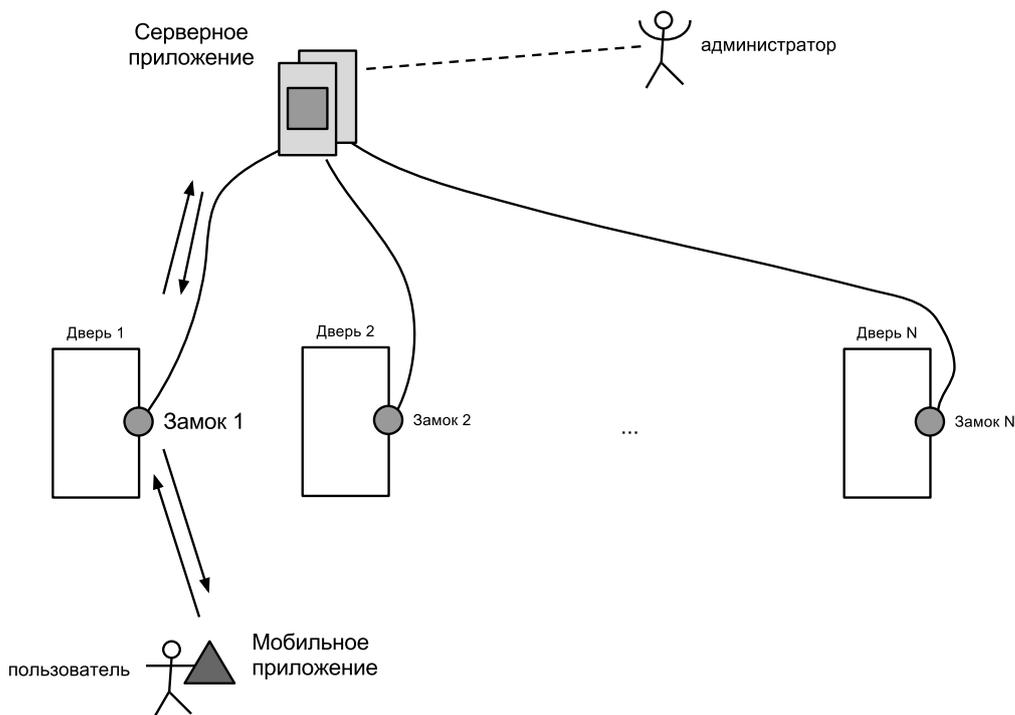


Рис. 1. Схема M-Block

2. Мобильное приложение M-Block устанавливает защищённое соединение с приложением замка M-Block.
3. Приложение замка передаёт мобильному приложению необходимую информацию для верификации (например, какие биометрические данные нужно получить от пользователя).
4. Пользователь с помощью мобильного приложения формирует необходимые биометрические данные для верификации (например, для голосовой биометрии — цифровую запись определённой фразы).
5. Мобильное приложение передаёт сформированные данные замку.
6. Приложение замка устанавливает защищённое соединение с сервером, передаёт ему биометрические данные и другую необходимую информацию.
7. Сервер M-Block запускает необходимый биометрический алгоритм, верифицирует и авторизует пользователя.
8. Сервер сообщает приложению замка результаты верификации и авторизации.
9. В случае положительного ответа приложение замка открывает дверь, в противном случае пользователю передаётся информация о причинах отказа.

Как и любая другая БСД, M-Block является весьма сложным программно-аппаратным комплексом. Более того, требования к ней могут меняться в процессе разработки, тестирования и апробации. Поэтому в первую очередь разработка системы должна начинаться с реализации рабочего прототипа, к которому предъявляется жёсткое условие возможности добавления и изменения требований в процессе разработки.

Целью данной статьи является обзор и выбор наиболее подходящего средства для реализации прототипа аппаратной и программной части замка.

## **1. Критерии оценки**

Для определения лучшего средства решения поставленной задачи необходимо сформулировать критерии оценки. Важно отметить, что перечисленные ниже критерии имеют значимость для реализации прототипа и могут быть неподходящими при выборе средств для реализации промышленной версии.

### **1.1. Взаимодействия с электромеханическим устройствами**

Для того, чтобы открывать и закрывать дверь, замок M-Block должен взаимодействовать с промышленным электромеханическим дверным замком. Поэтому устройство должно обязательно предоставлять программно-аппаратные возможности для управления электромеханическим приводом.

### **1.2. Скорость и удобство разработки**

Для прототипа большое значение имеет скорость и удобство разработки, поэтому устройство должно поддерживать программирование на высокоуровневом языке программирования с использованием удобной и эффективной IDE и других инструментов разработки.

### **1.3. Вычислительные ресурсы**

Приложение замка в ходе своей работы должно обмениваться сообщениями с мобильным приложением и сервером M-Block, обрабатывать и передавать значительные объёмы биометрических данных. Поэтому для прототипа замка ставятся следующие требования: частота процессора - не менее 128 МГц, размер оперативной памяти - не менее 100 МБ, размер внутренней памяти (или подключаемой) - не менее 100 МБ.

### **1.4. Беспроводное взаимодействие**

Замок должен устанавливать беспроводное защищённое соединение с мобильным приложением M-Block, поэтому прототип должен поддерживать как минимум один из современных способов беспроводной передачи данных на небольших расстояниях - Bluetooth или WiFi.

### **1.5. Взаимодействие по локальной сети**

Замок должен устанавливать защищённое соединение с сервером M-Block, поэтому прототип должен обеспечивать поддержку популярных стандартов локальных сетей Ethernet или WiFi.

### **1.6. Стек TCP/IP**

Исходя из поставленной задачи, прототип замка должен обеспечивать работу со стеком протоколов TCP/IP и предоставлять возможности реализации безопасного протокола на основе TCP и UDP сокетов. Также в качестве транспорта для протокола взаимодействия между замком и сервером может быть выбран HTTP, поэтому крайне желательна поддержка HTTP/HTTPS при программировании устройства замка.

### **1.7. Многозадачность**

Прототип замка должен одновременно поддерживать связь с мобильным приложением и сервером M-Block, поэтому решение для прототипа должно предоставлять средства для организации многозадачности.

### **1.8. Документация разработчика и поддержка**

Качество документации, наличие сторонних расширений, количество статей и различных учебных материалов в сети Интернет, активность сообщества разработчиков для каждого конкретного решения — все это является очень важным критерием для прототипа, так как минимизирует время на решение вопросов, возникающих в процессе разработки.

### **1.9. Стоимость**

Для прототипа замка стоимость не является ключевым фактором, так как прототип является уникальной и единичной реализацией. Однако данный параметр должен рассматриваться как дополнительный и значимый, в случае равенства в сравнении решений по вышеперечисленным критериям.

## **2. Существующие решения**

На данный момент существуют различные средства решения сформулированной задачи. В ходе подготовительной работы автором статьи было выделено подмножество предпочтительных решений, то есть заведомо более подходящих для поставленной задачи среди всего спектра решений. В их число вошли решения на базе Arduino, AllWinner A1X, Android и Raspberry Pi. В рамках данной статьи будут детально рассмотрены все предпочтительные решения и в результате сравнительного анализа по сформулированным критериям будет выявлено наиболее подходящее.

## 2.1. Arduino

Расширяемая аппаратно-вычислительная платформа, основными компонентами которой являются платы ввода/вывода, подключаемые шилды и среда разработки на языке Processing. Arduino предлагает различные модели плат, но все они основаны на микроконтроллере Atmel AVR, имея небольшие вычислительную мощность и размер по памяти [1].

Таблица 1. Arduino

Критерий	Комментарий	Оценка
Взаимодействие с электро-механическими устройствами	Arduino поддерживает прямое программирование портов GPIO, что является одной из его самых сильных сторон [2].	+
Скорость и удобство разработки	Язык Processing, а также IDE для программирования Arduino являются достаточно хорошими и проверенными решениями, но значительно проигрывают в скорости и удобстве разработки современным высокоуровневым языкам программирования и соответствующим им IDE. Также при использовании большого количества дополнительных шилдов может возникнуть серьёзная проблема нехватки GPIO разъёмов на плате и необходимости взаимоисключения шилдов.	+/-
Вычислительные ресурсы	Решения на базе Arduino имеют очень ограниченную и явно недостаточную для прототипа вычислительную мощность. Частота микроконтроллера не превышает 16 МГц, а размер оперативной памяти - 8 КБ [3].	-
Беспроводное взаимодействие	Arduino позволяет реализовать беспроводное взаимодействие по стандартам Bluetooth и WiFi за счет использования специальных плат или подключения дополнительных соответствующих шилдов. Так, например, взаимодействие по Bluetooth возможно за счёт использование платы Arduino BTArduinoBT или подключения шилда XBee [5].	+
Взаимодействие по локальной сети	Путем подключения дополнительных шилдов Ethernet [7] и WiFi [6] возможна реализация взаимодействия по соответствующему стандарту локальной сети. При этом плата Arduino может выступать как в роли клиента [9], так и сервера [8].	+

Критерий	Комментарий	Оценка
Стек TCP/IP	Стандартные библиотеки Arduino характеризуются очень низкоуровневым подходом к реализации взаимодействия по протоколам стека TCP/IP и отсутствием поддержки некоторых протоколов или определённых их возможностей. Например, Arduino не поддерживает взаимодействие по протоколу HTTPS по причине недостаточной вычислительной мощности.	–
Многозадачность	Arduino не имеет встроенных средств для обеспечения многозадачности. Поддержка многозадачности возможна только за счёт использования специальных сторонних библиотек или RTOS [10]. Однако оба варианта приведут к значительному усложнению реализации. В итоге многозадачность в Arduino является весьма проблемным вопросом с большим количеством «подводных камней».	–
Документация разработчика и поддержка	Официальный сайт Arduino предоставляет большое количество обучающих материалов и хорошую документацию разработчика. Также вокруг платформы существует большое сообщество активных разработчиков, что значительно облегчает процесс решения сложных вопросов, возникающих в процессе разработки.	+
Стоимость	80€ [11]	

## 2.2. Raspberry Pi

Одноплатный компьютер размером с кредитную карту, имеющий аппаратные характеристики, достаточные для комфортной работы современных десктопных и мобильных операционных систем. За счёт наличия 26 портов GPIO и сигнального процессора Raspberry Pi представляет хорошие возможности для разработки на его базе различных программно-аппаратных решений [12].

Таблица 2. Raspberry Pi

Критерий	Комментарий	Оценка
Взаимодействие с электромеханическими устройствами	Raspberry Pi имеет 26 портов GPIO и поддерживает их прямое программирование [13]. Также стоит отметить факт существования проекта raspberry-gpio-python, позволяющего управлять GPIO на высокоуровневом языке Python [14].	+

Критерий	Комментарий	Оценка
Скорость и удобство разработки	Официально поддерживаемым языком для разработки приложений на Raspberry Pi является Python [15], для которого существует большое количество различных IDE. Оба эти фактора обеспечивают хорошие возможности для скоростной и комфортной разработки приложений под Raspberry Pi.	+
Вычислительные ресурсы	Raspberry обладает достаточными физическими характеристиками, имея 700 МГц процессор, 256 или 512 МБ памяти [15] и возможность подключения до 32 ГБ внешней памяти.	+
Беспроводное взаимодействие	Реализация взаимодействия по Bluetooth и WiFi возможна путём подключения к USB порту Raspberry Pi WiFi/Bluetooth адаптера.	+
Взаимодействие по локальной сети	Raspberry Pi предоставляет встроенные средства для поддержки стандарта Ethernet и за счёт расширения позволяет обеспечить поддержку WiFi. При этом Raspberry Pi может выступать как в роли клиента, так и в роли сервера.	+
Стек TCP/IP	Стандартной операционной системой Raspberry Pi является Raspbian [16] — оптимизированная и преднастроенная версия Debian. Raspbian содержит в себе полноценную реализацию стека TCP/IP, предоставляя необходимые средства для создания TCP и UDP сокетов, а также взаимодействия по протоколам HTTP и HTTPS.	+
Многозадачность	Raspbian, также как и Debian, является многозадачной операционной системой.	+
Документация разработчика и поддержка	Для Raspberry Pi в сети Интернет можно найти большое количество документации и различных учебных материалов. Также вокруг Raspberry Pi сформировалось большое и очень активное сообщество разработчиков, которое активно развивает платформу, каждый день появляются новые статьи о решении самых различных задач с помощью Raspberry Pi. Все это позволяет очень быстро найти ответ на различные вопросы, возникающие в процессе разработки для Raspberry Pi.	+
Стоимость	21.60€ [17]	

### 2.3. Android

Открытая мобильная операционная система, основанная на ядре Linux. На данный момент на рынке представлено огромное количество различных устройств под управлением Android, в том числе смартфонов, планшетов, электронных книг, домашних мультимедиа центров, фотоаппаратов и телевизоров [18].

Для решения сформулированной задачи наиболее удачным форм-фактором является смартфон, так как при прочих равных он имеет компактный размер, необходимые аппаратные возможности и наиболее качественную документацию. Поэтому оценка Android будет даваться именно на основе возможностей Android смартфонов.

Таблица 3. Android

Критерий	Комментарий	Оценка
Взаимодействие с электромеханическими устройствами	Существующие Android смартфоны не имеют GPIO портов и возможностей для прямого взаимодействия с электромеханическими устройствами [19]. Тем не менее, Android предоставляет два обходных решения. Первое заключается в поддержке взаимодействия с внешними устройствами по USB [20], что подходит не для всех устройств и внесёт дополнительную сложность в реализацию прототипа. Вторым вариантом является взаимодействие в рамках подхода Android ADK [20], при этом в дополнении к устройству на базе Android для реализации прототипа замка необходимо использовать дополнительное решение, например, плату Arduino ADK.	+/-
Скорость и удобство разработки	Стандартным языком для разработки приложений для Android является язык Java, а стандартной IDE — Eclipse [18]. Эти два фактора обеспечивают хорошие возможности для скоростной и комфортной разработки Android приложений.	+
Вычислительные ресурсы	Практически все современные Android смартфоны соответствуют необходимым физическим требованиям [19] в силу более высоких минимальных требований для работы ОС Android [21].	+
Беспроводное взаимодействие	Практически все современные смартфоны на Android содержат WiFi и Bluetooth модули. [19].	+
Взаимодействие по локальной сети	Практически все современные смартфоны на Android содержат WiFi модуль и не содержат порт Ethernet [19].	+

Критерий	Комментарий	Оценка
Стек TCP/IP	ОС Android содержит полноценную реализацию стека TCP/IP, а Android SDK предоставляет все необходимые средства для создания TCP и UDP сокетов и взаимодействия по протоколам HTTP и HTTPS [23].	+
Многозадачность	Android является многозадачной операционной системой, также предоставляя стандартные средства для разработки многопоточных приложений [24].	+
Документация разработчика и поддержка	Официальный сайт Android Developers содержит исчерпывающее количество документации и учебных материалов по разработке приложений на базе Android SDK. Также на тему разработки приложений для Android было сделано большое количество докладов на различных конференциях, написано огромное количество статей и обсуждений на различных форумах.	+
Стоимость	От \$160 [25]	

#### 2.4. AllWinner A1X

Семейство одноядерных однокристальных систем. На данный момент представлено решениями A10, A13, A10s и A31. Содержит одно ядро ARM Cortex-A8 как основной процессор CPU и графический процессор GPU Mali 400 [26].

Самого по себе AllWinner A1X недостаточно для решения поставленной задачи, однако на его базе существует ряд интересных решений: BeagleBone [28], Hackberry [30], Cubieboard [27], Gooseberry [29] и ряд других. Для поставленной задачи наиболее подходящими вариантами являются решения, содержащие порты GPIO и доступные для заказа — это CubieBoard и BeagleBone. Данные устройства весьма схожи между собой по техническим характеристикам и были разработаны позже Raspberry Pi с целью составить достойную конкуренцию за счет более высоких аппаратно-вычислительных возможностей при сравнимой с Pi цене. Поэтому оценка AllWinner A1X будет даваться именно на основе возможностей CubieBoard и BeagleBone.

Таблица 4. AllWinner A1X

Критерий	Комментарий	BeagleBone	CubieBoard
Взаимодействие с электро-механическими устройствами	BeagleBone и CubieBoard поддерживают прямое программирование портов GPIO. Для BeagleBone это возможно по умолчанию средствами языка Bash [31], для CubieBoard необходимо поставить Linux в качестве ОС.	+	+/-
Скорость и удобство разработки	Предустановленной ОС для CubieBoard является Android, программирование которого возможно на Java в Eclipse IDE. Для BeagleBone предустановленной ОС является Angstrom [32], программирование которого возможно на JavaScript в Cloud9 IDE [28]. При установке Linux на CubieBoard для написания программ можно использовать C/C++ и Eclipse IDE [27]. Все перечисленные варианты должны обеспечить достаточно высокую скорость и удобство разработки.	+	+
Вычислительные ресурсы	CubieBoard и BeagleBone обладают достаточными физическими характеристиками. BeagleBone: 720 МГц процессор, 256 МБ RAM, возможность подключения до 32 ГБ внешней памяти [28]. CubieBoard: 1 ГГц процессор, 1 ГБ RAM, 4 ГБ внутренней NAND памяти и возможность подключения до 32 ГБ внешней памяти [27].	+	+
Беспроводное взаимодействие	Реализация взаимодействия по Bluetooth и WiFi теоретически возможна путем подключения к USB порту соответствующих адаптеров, однако на практике сообщается о проблемах и необходимости внесения патчей в ядро системы.	+/-	+/-
Взаимодействие по локальной сети	BeagleBone и CubieBoard имеют встроенный Ethernet порт и все необходимые средства для написания как клиентских, так и серверных приложений.	+	+

Критерий	Комментарий	BeagleBone	CubieBoard
Стек TCP/IP	Angstrom, Android, Ubuntu, как и все другие популярные варианты ОС для CubieBoard и BeagleBone, имеют полноценную реализацию стека TCP/IP и предоставляют необходимые для прототипа сетевые возможности.	+	+
Многозадачность	Angstrom, Android, Ubuntu как и все другие популярные варианты ОС для CubieBoard и BeagleBone являются многозадачными операционными системами.	+	+
Документация разработчика и поддержка	CubieBoard и BeagleBone значительно проигрывают Raspberry Pi в качестве документации разработчика, по количеству статей и обсуждений, в размере и активности сообщества разработчиков, по наличию сторонних расширений и приложений, а также другим смежным параметрам. Например, на первой странице выдачи Google по запросу «raspberrypi gpio» можно найти ссылку на специально созданную для этого библиотеку <code>raspberrypi-gpio-python</code> [14] и ещё 8 обучающих статей по этому вопросу. Запрос «beaglebone gpio» — в выдаче библиотеки <code>beaglebone-gpio</code> для языка C и 4 статьи. Запрос «cubieboard gpio» — в выдаче только спецификации <code>cubieboard</code> и безуспешные обсуждения на форумах.	+/-	-
Стоимость	CubieBoard — \$60 [34]. BeagleBone — \$89 [33]		

### 3. Заключение

Из сравнительного анализа и оценок для каждого решения следует:

1. Наиболее подходящим средством для реализации прототипа замка M-Block является Raspberry Pi, так как данное решение получило положительные оценки для всех критериев, не получив ни одной негативной оценки.
2. Arduino не может рассматриваться как возможный вариант для решения,

так как имеет несоответствия по критериям «Вычислительные ресурсы» и «Стек TCP/IP».

3. Android и AllWinner A1X стоит признать подходящими и весьма хорошими вариантами для прототипа, но проигрывающими в сравнении с Raspberry Pi. Android — из-за не возможности прямого программирования GPIO портов, а решения на базе AllWinner A1X — из-за недостаточного качества документации и низкой активности сообщества.

## ЛИТЕРАТУРА

1. Официальный сайт Arduino [Электронный ресурс]. URL: <http://www.arduino.cc/> (дата обращения: 31.03.2013).
2. Arduino Analog Input Pins // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Tutorial/AnalogInputPins> (дата обращения: 31.03.2013).
3. Arduino Memory // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Tutorial/Memory> (дата обращения: 31.03.2013).
4. Arduino Bluetooth Shield // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Main/ArduinoBoardBluetooth> (дата обращения: 31.03.2013).
5. XBee Shield Wiki // Seeed Wiki [Электронный ресурс]. URL: [http://www.seeedstudio.com/wiki/XBee\%C2\%AE\\_Shield](http://www.seeedstudio.com/wiki/XBee\%C2\%AE_Shield) (дата обращения: 31.03.2013).
6. Arduino WiFi Shield // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Main/ArduinoWiFiShield> (дата обращения: 31.03.2013).
7. Arduino Ethernet Shield // Официальный сайт Arduino. URL: <http://arduino.cc/en/Main/ArduinoEthernetShield> (дата обращения: 31.03.2013).
8. WebServer // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Tutorial/WebClient> (дата обращения: 31.03.2013).
9. WiFi Twitter Client // Официальный сайт Arduino [Электронный ресурс]. URL: <http://arduino.cc/en/Tutorial/WiFiTwitterClient> (дата обращения: 31.03.2013).
10. RTOS for Arduino // Out There [Электронный ресурс]. URL: <http://arduino.cc/en/Tutorial/WiFiTwitterClient> (дата обращения: 31.03.2013).
11. Arduino Starter Kit // Arduino Store [Электронный ресурс]. URL: [http://store.arduino.cc/ww/index.php?main\\_page=product\\_info&cPath=2\\_23&products\\_id=185](http://store.arduino.cc/ww/index.php?main_page=product_info&cPath=2_23&products_id=185) (дата обращения: 31.03.2013).
12. Официальный сайт Raspberry Pi [Электронный ресурс]. URL: <http://www.raspberrypi.org/> (дата обращения: 31.03.2013).
13. RPi Hardware // eLinux.org [Электронный ресурс]. URL: [http://elinux.org/RPi\\_Hardware](http://elinux.org/RPi_Hardware) (дата обращения: 31.03.2013).
14. Python library for GPIO access on Raspberry Pi // Google Code [Электронный ресурс]. URL: <https://code.google.com/p/raspberry-gpio-python/> (дата обращения: 31.03.2013).

15. FAQs // Официальный сайт Raspberry Pi [Электронный ресурс]. URL: <http://www.raspberrypi.org/faqs> (дата обращения: 31.03.2013).
16. Официальный сайт Raspbian [Электронный ресурс]. URL: <http://www.raspbian.org/> (дата обращения: 31.03.2013).
17. Raspberry Pi // RS Online [Электронный ресурс]. URL: <http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi> (дата обращения: 31.03.2013).
18. Официальный сайт Android Developers [Электронный ресурс]. URL: <http://developers.android.com> (дата обращения: 31.03.2013).
19. Сравнение Android устройств // Википедия [Электронный ресурс]. URL: [http://ru.wikipedia.org/wiki/Сравнение\\_устройств\\_с\\_Android](http://ru.wikipedia.org/wiki/Сравнение_устройств_с_Android) (дата обращения: 31.03.2013).
20. USB Host and Accessory // Android Developers [Электронный ресурс]. URL: <http://developer.android.com/guide/topics/connectivity/usb/index.html> (дата обращения: 31.03.2013).
21. Android Hardware Requirments // Android Hardwares [Электронный ресурс]. URL: <http://androidhardwares.com/android-hardware-development/android-hardware-requirements/> (дата обращения: 31.03.2013).
22. Performing Network Operations // Android Developers [Электронный ресурс]. URL: <http://developer.android.com/training/basics/network-ops/index.html> (дата обращения: 31.03.2013).
23. Performing Network Operations // Android Developers [Электронный ресурс]. URL: <http://developer.android.com/training/basics/network-ops/index.html> (дата обращения: 31.03.2013).
24. Painless Threading // Android Developers [Электронный ресурс]. URL: <http://android-developers.blogspot.com/2009/05/painless-threading.html> (дата обращения: 31.03.2013).
25. Самый дешевый в мире Android смартфон // АйМобилка [Электронный ресурс]. URL: <http://icellphone.ru/kitajcy-pokazali-samyj-deshevyj-v-mire-android-smartfon.html> (дата обращения: 31.03.2013).
26. AllWinner A Serial // Официальный сайт AllWinner Technology [Электронный ресурс]. URL: <http://www.allwinnertech.com/en/product/A-Serial.html> (дата обращения: 31.03.2013).
27. Официальный сайт Cubieboard [Электронный ресурс]. URL: <http://linux-sunxi.org/Cubieboard> (дата обращения: 31.03.2013).
28. Официальный сайт GooseBerry [Электронный ресурс]. URL: <http://beagleboard.org/bone> (дата обращения: 31.03.2013).
29. Официальный сайт Gooseberry [Электронный ресурс]. URL: <http://gooseberry.atSPACE.co.uk/> (дата обращения: 31.03.2013).
30. Официальный сайт Hackberry [Электронный ресурс]. URL: <https://www.miniand.com/products/Hackberry%20A10%20Developer%20Board> (дата обращения: 31.03.2013).
31. Accessing GPIO // BeagleBone for Dummies [Электронный ресурс]. URL: <http://bbfordummies.blogspot.ru/2009/07/1.html> (дата обращения: 31.03.2013).
32. Angstrom Distribution // BeagleBoard.org [Электронный ресурс]. URL: <http://>

- beagleboard.org/project/angstrom/ (дата обращения: 31.03.2013).
33. BeagleBone A6 // Adafruit [Электронный ресурс]. URL: <https://www.adafruit.com/products/513> (дата обращения: 31.03.2013).
34. CubieBoard // DFRobot [Электронный ресурс]. URL: [http://www.dfrobot.com/index.php?route=product/product&filter\\_name=cubieboard&product\\_id=881#.UVxjTZMqwUR](http://www.dfrobot.com/index.php?route=product/product&filter_name=cubieboard&product_id=881#.UVxjTZMqwUR) (дата обращения: 31.03.2013).

## **БИХЕВИОРИСТИЧЕСКАЯ ИДЕНТИФИКАЦИЯ ПРОЦЕССОВ**

**Р.С. Прохоров**

В данной статье рассмотрен подход, позволяющий осуществить идентификацию процесса по его поведению. Подход основан на преобразовании потока событий в поток контейнеров событий и использовании такого потока для подачи на вход интеллектуального анализатора, построенного с помощью мультинейронной сети. В качестве «ядра» анализатора используется принципиально новая структура композитной нейронной сети, названная мультинейронной сетью. Подробно описаны варианты используемых преобразований, а также сами используемые преобразования.

### **Введение**

Проблема идентификации процессов возникает в широком классе задач защиты информации, таких как борьба с вредоносным программным кодом, построение замкнутой программной среды, политика ограниченного использования программ и другие. На сегодняшний день наибольшее распространение получили подходы, основанные на анализе исполняемого файла, либо анализе непосредственно исполняемого кода [1]. Идентификация процесса может производиться по имени исполняемого файла, хэш-значению исполняемого кода или сигнатуре исполняемого кода. Все эти методы могут быть преодолены злоумышленником с помощью преобразований исполняемого кода, не влияющих на его функциональность. Несмотря на то, что разрабатываются все новые и новые алгоритмы сигнатурного анализа [2–4], всегда можно создать специальный алгоритм мутации, обходящий свежие системы защиты. В связи с этим остаётся актуальной задача разработки технологии идентификации процесса по некоторому набору признаков.

Как легко понять, проблема идентификации процессов относится к задачам распознавания образов и к ней могут быть применены соответствующие технологии искусственного интеллекта. Главная проблема, с которой приходится сталкиваться, состоит в формировании «образа» процесса, который, с одной стороны, однозначно его характеризует, а с другой стороны устойчив к воздействиям исполняемого кода, не изменяющим функционал процесса.

Бихевиористический анализ является частью иммунной системы, построенной в работе [5], где он дополняется сигнатурным анализом. В работе [6] предлагается архитектура нейросетевой системы анализа поведения процессов операционной системы реального времени, используемых в специализированных встраиваемых системах. В работах [7, 8] описано формирование образа поведения процесса по особой структуре, названного свойством (feature). Принцип этого формирования заключается в повторении присущих процессу свойств.

Идея, лежащая в основе бихевиористического анализа: субъект может выполнять только такие действия, которые ему предписаны. Значит, имея набор событий, связанных с каким-либо субъектом, мы можем попытаться определить, чем является этот субъект.

Следует отметить, что задача идентификации процесса является задачей распознавания образов. Здесь можно выделить два типа образов: статический и динамический. Статический образ имеет все признаки «здесь и сейчас», ему свойственна полнота набора признаков — исследуемая сущность не может измениться, в противном случае это будет уже другая сущность. Например, задача распознавания образов на картинке. Динамический же образ растянут во времени, ему свойственна потоковая природа формирования — рождение в процессе.

Таким образом, необходимо учитывать особенности формирования образа субъекта. Такой образ также будет устойчив к нефункциональному изменению тела субъекта — изменению пассивной части, то есть части, не влияющей на поведение объекта.

Для дальнейшего изложения необходимо выделить несколько основных понятий.

*Образом* субъекта назовём след, оставленный его поведением в среде, для процесса — это события в компьютерной системе.

*Среда* — сущность, на которую воздействует субъект, которая преобразуется под его влиянием, для процесса — операционная система.

*Поведение* — это поток действий, для процесса — поток событий, которые произошли под влиянием данного процесса.

## 1. Анализ бихевиористических характеристик

Будем считать поведение — потоком событий, имеющим неопределённую длину. Пусть  $E$  (events) — множество всевозможных событий, которые могут произойти в рассматриваемой среде. Тогда для каждого субъекта, функционирующего в данной среде за время  $\Delta t$ , можно определить его поведение  $b = (e_1, e_2, \dots, e_n)$ , где  $e \in E$ .

Для каждого события, произошедшего в среде, возможно определить его тип, субъект, который его породил, а также некоторые данные, присущие конкретному типу событий. Поток событий разделяется на группы так, что каждой группе соответствует субъект, породивший все события из данной группы. Необходимо присвоить каждому субъекту архетип, анализируя только поток

связанных с ним событий. В результате это сводится к задаче классификации с динамическим формированием образов субъектов.

В рассматриваемых средах функционирует некоторое конечное число субъектов. Каждый из них может породить только подмножество множества всех возможных событий, обозначим его  $\bar{E}$ . Тогда по мощности данного множества возможно следующее разделение сред: тривиальные, для которых  $|\bar{E}| = 1$ , квазидетерминированные, реальные, свободные, хаотичные, где для каждого процесса  $\bar{E} = E$ .

Задача идентификации субъекта в тривиальной системе имеет очевидное решение: взаимоднозначное отображение множества событий во множество архетипов субъектов. Для хаотичной системы — наоборот. Задача неразрешима, поскольку любой субъект может породить любое событие.

Квазидетерминированной системой назовём среду, в которой субъект генерирует хотя бы по одному событию из большинства присущих ему классов. Реальные среды — среды, в которых субъект не может раскрыть своё поведение за один интервал времени. Свободной средой назовём среду с изменяющимся субъектом. В данной статье рассматривается функционирование субъектов в *реальных* средах.

## 2. Формирование бихевиористического образа процесса

Рассмотрим подробнее поток событий, ассоциированный с субъектом. Пусть  $a = (a_1, a_2, \dots, a_n)$  — последовательность событий, где  $a_i$  — действие субъекта, причём для каждого  $a_i$  существует класс события  $e \in E$ . Назовём такую последовательность *поведением субъекта* или *поведением*. Поведение субъекта за всё время слежения назовём *глобальным*. Любую её подпоследовательность назовём *локальным* поведением.

Для формирования образа субъекта используется функция разбиения глобального поведения на набор локальных поведений. Этот набор обладает рядом следующих свойств:

1. События внутри подмножеств сохраняют порядок (поведение — последовательность событий).
2. Объединение всех локальных поведений есть глобальное поведение.
3. Одно событие может принадлежать только одному локальному поведению из всего набора.

Из данных свойств следует, что ни одно событие не будет пропущено, что позволило наиболее точно сформировать образ исследуемого субъекта у анализатора. Таким образом, с каждым новым локальным поведением, поступающим на вход анализатора, формируется образ субъекта.

### 3. Идентификация процесса по бихевиористическому образу

Ядром системы бихевиористического анализа является *анализатор*. Задача обработчика — определять, имеет ли субъект конкретный архетип. Такая реализация схожа с реальной иммунной системой человека, где каждое анти-тело реагирует только на антигены, которые попадают в радиус стимуляции. В данной статье описан детектор, который умеет отличать только антигены конкретного архетипа.

Если представить анализатор в виде «чёрного ящика», то можно определить входы и выходы.

*Вход* — последовательность событий за некий промежуток времени (локальное поведение).

*Выход* — имеет ли процесс искомый архетип или нет.

Следует отметить, что внутри такого «чёрного ящика» может быть сколь угодно сложная интеллектуальная структура.

Каждое локальное поведение преобразуется с помощью следующей функции:

$$f : B \rightarrow V. \quad (1)$$

Здесь  $B$  — множество локальных поведений, а  $V$  — множество входных векторов. Большинство интеллектуальных структур принимают на вход вектор в том или ином виде, поэтому и был выбран данный интерфейс. В качестве функции, осуществляющей преобразование (1), выбран *количественный показатель*. Для этого использован вектор длиной  $m = |E|$ : каждый из элементов соответствует одному классу событий, отражая количество встреченных событий такого класса в данном контейнере.

В реальных средах, особенно компьютерных системах, субъектам свойственно порождать длинные последовательности повторяющихся событий, что при использовании количественного показателя может вернуть входной вектор с большим числом нулевых значений, например  $(0, 0, 0, 0, 18, 0, 0, 0, 0, 1, 0, 0)$ . Как известно, эффективность любой нейронной сети стремительно падает при работе с такими входными векторами: результат просто «забывается» нулевыми значениями и получается, что выходной вектор практически не несёт полезной информации. Данная особенность сильно затрудняет традиционное использование нейронных сетей при таком подходе в данной задаче.

Для предотвращения такого эффекта необходимо сокращать число входов и обрабатывать только ненулевые части. Были рассмотрены следующие методы: группировка, трансляция и фильтрация.

*Группировка* — метод, при котором входной вектор разбивается на несколько векторов меньшего размера так, что сумма их длин остаётся равной длине первоначального вектора. Далее необходимо разделить множество событий на группы, соответствующие определённым классам действий. Сгруппировав их, получим объединение непересекающихся множеств.

*Трансляция* заключается в формировании множества надклассов событий. Каждый класс событий имеет некий образ — надкласс. Было использовано

следующее сюръективное отображение:

$$f : E \rightarrow S. \quad (2)$$

Здесь  $S$  — множество обобщённых или укрупнённых классов событий.

*Фильтрация* — данный метод «отсеивает» лишние классы событий, чтобы сконцентрироваться на наиболее значимых из них. Было установлено, что данный метод в решаемой задаче будет искажать результат — нельзя игнорировать какие-либо классы событий, потому что субъект, который порождает только такие события, будет иметь нулевой архетип.

Самым эффективным методом является группировка, поскольку она может сократить число входов на порядок. Очевидно, что для группировки необходимо использование нескольких дочерних обработчиков, каждый из которых работает с небольшой группой классов событий. В результате мы имеем вектор ответов «да/нет». Для получения финального вектора необходимо разработать особую интеллектуальную структуру с подуровнями, которая способна принимать финальное решение. Назовём такую структуру *мультинейронной сетью* — сетью, состоящей из множества «метанейронов» — других нейронных сетей. Для дополнительного сокращения размерности вектора была добавлена трансляция параметров внутри каждой группы. В методах сокращения числа входов была использована *частичная фильтрация*: был выделен дополнительный надкласс «другие события», куда отображались малозначимые или схожие события.

#### 4. Мультинейронная сеть

В современном мире существует масса примеров, когда какую-то сущность можно рассмотреть как набор таких же сущностей. Она может быть представлена в виде этого набора или быть этим набором. Такой подход при рассмотрении того или иного объекта применяется обычно для того, чтобы ограничить охватываемую область. Это очень сильно способствует пониманию, как работает объект. Идея взаимодействия схожих сущностей и «советников» была заимствована у Кеннеди и Эберхарта [9].

*Интеллектуальный анализатор* был разбит на малые анализаторы, причём каждый из них является таким же самостоятельным анализатором, но отвечающим на вопросы более узкого профиля. Также был установлен интерфейс каждого анализатора. Далее, для каждой группы анализаторов сформирован модуль разбиения входов и модуль согласования выходов.

*Модуль разбиения входа* отвечает за преобразование, разбивающее исходный вектор на множество векторов, что является функцией преобразования множества входных векторов во множество наборов входных векторов меньшей размерности  $f : V \rightarrow \bar{V}^n$ , где  $n$  — число нейронных подсетей.

*Модуль согласования выхода* отвечает за преобразование  $f : P \rightarrow \bar{P}^n$ , где  $n$  — число компонентов, а  $P$  — множество выходных векторов. Такой модуль формирует финальный результат. Следует отметить, что каждый из компонентов может являться как простой нейронной сетью, так и мультинейронной.

Мультинейронная сеть была реализована на основе нижеописанных компонентов.

*Атомарная нейронная сеть  $N$*  — простая неделимая нейронная сеть, то есть не являющаяся мультинейронной, на  $N$  входов. Каждая из таких сетей имеет два выхода.

*Модуль разбиения входа* преобразует одиночный контейнер в набор пар «группа классов — входной вектор».

*Транслятор надклассов событий* преобразует конкретный класс события в укрупнённый надкласс событий. Выполняет отображение (2).

*Индексное хранилище нейронных подсетей* содержит набор пар «группа классов — нейронная подсеть». Для обеспечения быстрого действия используется индекс, построенный на основе бинарного дерева сортировки. С помощью индекса осуществляется быстрый поиск нужной подсети. Время поиска не превышает  $\log N$ .

*Индексное хранилище учителей для нейронных подсетей* содержит набор пар «группа классов — учитель нейронной подсети». Для обеспечения быстрого действия также используется индекс, построенный на основе бинарного дерева сортировки. С помощью индекса осуществляется быстрый поиск нужного учителя. Время поиска не превышает  $\log N$ .

*Модуль согласования выхода* содержит информацию об успешности каждой нейронной подсети: процент правильно определённых процессов. Принимает на вход вектор ответов «да/нет». Выносит финальный вердикт о породившем контейнер субъекте. Использует схему «уровень доверия подсети», чтобы получить финальный результат.

*Функция уровня доверия подсети.* Пусть мы имеем  $n$  двумерных векторов  $P$ . Представим их как два вектора  $R_1$  и  $R_2$ .  $R_i = (p_{i_1}, p_{i_2}, \dots, p_{i_n})$

Рассмотрим *вектор доверия*  $T = (t_1, t_2, t_n)$ . Данный вектор формируется в процессе тестирования сети, после её обучения. Тогда финальный результат вычисляется по формуле

$$Result = \begin{cases} yes, & res > 0; \\ no, & res < 0; \\ not\ defined, & res = 0; \end{cases}$$

где  $res = (R_1, T) - (R_2, T)$ .

Вектор  $T$  формируется следующим образом. Каждый раз, когда конкретная подсеть даёт результат, и мы можем определить, правильный ли он или нет, в реестр успехов вносится соответствующая запись:

1. Для каждой подсети хранятся два значения: число успешных идентификаций  $s$  и общее число идентификаций  $n$ .
2. Если идентификация была проведена верно, то для подсети, которая провела эту идентификацию, на единицу увеличиваются значения  $s$  и  $n$ .
3. Если подсеть ошиблась, то для неё увеличивается общее число идентификаций ( $n$ ).

В любое время мы можем получить уровень доверия конкретной нейронной подсети:

$$t_i = \begin{cases} \frac{s_i}{n_i}, & n_i > 0; \\ 0, & n_i = 0. \end{cases}$$

Для формирования вектора уровня доверия необходимо два этапа обучения:

1. Обучение нейронной сети. На этом этапе формируются весовые коэффициенты самих нейронных подсетей.
2. Тестирование нейронной сети. Сеть тестируется и каждому компоненту (подсети) выставляется соответствующее значение уровня доверия.

Все вышеописанные компоненты объединяются в анализатор. Обучение мультинейронной сети происходит следующим образом:

1. Входной вектор разбивается на набор  $n$  малых входных векторов с помощью модуля разбиения выхода.
2. Для каждого малого вектора события, содержащиеся в нем, транслируются в события укрупнённых классов с помощью транслятора надклассов событий.
3. Каждый из малых входных векторов подаётся на вход соответствующему учителю нейронной сети из индексного хранилища.

Вычисление результата с помощью обученной сети:

1. Входной вектор разбивается на набор  $n$  малых входных векторов с помощью модуля разбиения выхода.
2. Для каждого малого вектора события, содержащиеся в нем, транслируются в события укрупнённых классов с помощью транслятора надклассов событий.
3. Каждый из малых входных векторов подаётся на вход соответствующей нейронной подсети из индексного хранилища.
4. Каждая нейронная подсеть вычисляет результат.
5. Набор  $n$  ответов нейронных подсетей подаётся на вход модулю согласования выхода.
6. Модуль согласования выхода выдаёт финальный результат.

Таблица 1. Соотношение «сырых» и укрупнённых классов событий по группам

Название	Классы событий ( $C_i$ )	Классы событий ( $S_i$ )
Дисковые операции	11	3
Сетевая активность	26	5
Файловая активность	18	5
Операции с реестром	19	4

## 5. Использование мультинейронной сети

Как отмечалось, данная сеть лишена недостатка множества лишних нулевых параметров, но нуждается в более тонкой настройке, а именно, формировании групп параметров. Эффективность действия мультинейронной сети напрямую зависит от удачного распределения параметров по группам.

Было определено входное преобразование контейнера во множество малых векторов. Некоторые эффективные методы отбора данных представлены в работах [10, 11]. В данной статье, как отмечалось выше, предлагается использование группировки, трансляции и частичной фильтрации для формирования таких множеств.

*Группировка.* Основные рассматриваемые операции процессов поделены на четыре группы классов: дисковые операции, сетевая активность процесса, файловая активность и операции с реестром. Классы событий, составляющие каждую из этих групп, будут подробно рассмотрены далее. Для фиксирования событий в операционной системе Windows 7 была использована библиотека с открытым исходным кодом Event Tracing for Windows.

*Трансляция.* В каждой из групп классов событий было выделено до пяти надклассов.

*Частичная фильтрация.* Частичная фильтрация проявляется в добавлении «неважных» надклассов событий в каждую группу.

На основе вышеописанного построена таблица 1.

Таким образом, получено четыре трёхслойных (с одним скрытым слоем) нейронных подсети: персептроны  $3 - 3 - 2$ ,  $5 - 5 - 2$ ,  $5 - 5 - 2$  и  $4 - 4 - 2$ . В качестве функции активации используется сигмовидная функция. Каждая из них имеет два выхода: уверенность в ответе «да» и уверенность в ответе «нет».

## 6. Результаты

В таблице 2 показана эффективность применяемого метода к некоторым процессам после тестирования и после проверки на финальном множестве. Как показывают результаты исследования, данный подход крайне эффективен для «редких» процессов со специфическим поведением и средне эффективен для «популярных» процессов с довольно широким спектром порождаемых событий.

Таблица 2. Соотношение «сырых» и укрупнённых классов событий по группам

Процесс	Тестирование			Проверка		
	Успешность		%	Успешность		%
explorer	4037	6219	64,91%	4105	6206	66,15%
System	6215	6318	98,37%	5006	6300	79,46%
chrome	4820	6348	75,93%	5068	6201	81,73%
svchost	5636	6178	91,23%	5726	6309	90,76%
ekrn	5717	6215	91,99%	5805	6355	91,35%
wmagent	5755	6274	91,73%	5676	6147	92,34%
vmware-authd	5928	6234	95,09%	6032	6309	95,61%
PDFCreator	5950	6207	95,86%	6024	6285	95,85%
Dropbox	6035	6193	97,45%	6124	6277	97,56%
Skype	6091	6233	97,72%	6198	6329	97,93%
BisonHK	6130	6231	98,38%	6145	6267	98,05%
audiodg	6229	6285	99,11%	6244	6293	99,22%
QipGuard	6197	6236	99,37%	6280	6311	99,51%
wmpnetwk	6299	6310	99,83%	6119	6128	99,85%
SynTPEnh	6264	6271	99,89%	6278	6287	99,86%
services	6208	6214	99,90%	6194	6202	99,87%
NeuroIncinerateGUI	6477	6487	99,85%	6135	6142	99,89%
PrnStatusMX	6251	6255	99,94%	6220	6227	99,89%
VCSExpress	6288	6294	99,90%	6262	6267	99,92%

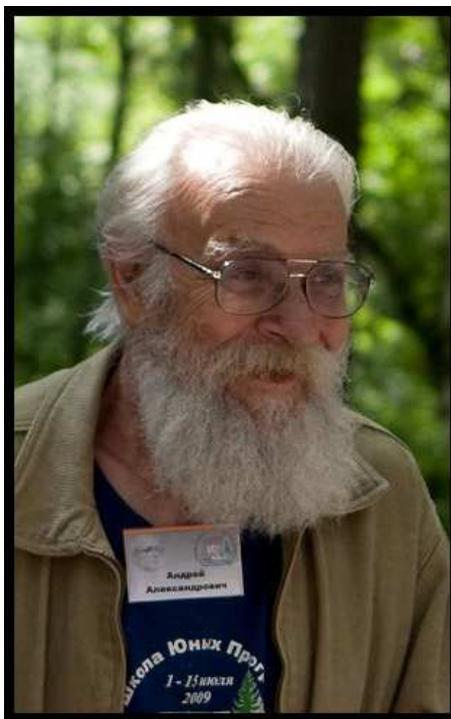
## Выводы

В данной статье был рассмотрен подход, позволяющий осуществить идентификацию процесса по его поведению. Эффективность колеблется в районе 0,66–0,99. Так как вредоносное программное обеспечение зачастую является сугубо специфичным, то такая система должна достаточно эффективно определять архетип данных процессов. Преодолена проблема входных векторов для нейронных сетей, почти полностью заполненных нулями, с помощью мульти-нейронной сети.

## ЛИТЕРАТУРА

1. Булахов Н.Г., Калайда В.Т. Методы обнаружения и обезвреживания саморазмножающихся вирусов // Доклады ТУСУРа. Июнь 2008. № 2(18). Ч. 1. С. 78–82.
2. Christodorescu M., Jha S., Seshia S., Song D., Bryant R. Semantics-aware malware detection // The IEEE Symposium on Security and Privacy. 2005.
3. Jha M.C. a. S. Static analysis of executables to detect malicious patterns // USENIX Security Symposium. 2003.
4. Dai R.G. a. J.L.J. Efficient Virus Detection Using Dynamic Instruction Sequences // Journal of Computers. 2009.
5. Ваганов М.Ю. Гибридная искусственная иммунная система защиты компьютера от процессов с аномальной активностью: диссертация кандидата технических наук: 05.13.19. Санкт-Петербург, 2012. 92 с.
6. Валеев С.С., Дьяконов М.Ю. Нейросетевая система анализа аномального поведения вычислительных процессов в микроядерной операционной системе // Вестник УГАТУ. 2012. Т. 14, № 5(40). С. 198–204.
7. Lo D., Cheng H., Han J., Khoo S., Sun C. Classification of Software Behaviors for Failure Detection: A Discriminative Pattern Mining Approach. // KDD'09, Июнь 28 – Июль 1, 2009. Париж, Франция.
8. Ahmadi M., Sami A., Rahimi H., Yadegari B. Iterative System Call Patterns Blow the Malware Cover // Security for The Next Generation. 2011.
9. Kennedy J., Eberhart R.C. Swarm Intelligence. Academic Press, 2001. 512 с.
10. Boursard H., Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition // Biological Cybernetics. 1988. N. 59. P. 291–294.
11. Oja E. A simplified neuron model as a principal component analyzer // Journal of Mathematical Biology. 1982. N. 15. P. 267–273.

## Андрей Александрович Берс



**1934-2013**

В Новосибирске 28 января 2013 года скоропостижно скончался Андрей Александрович Берс, наш коллега, член редколлегии нашего журнала. Андрей Александрович был одним из пионеров советской кибернетики и программирования. Умер на 79 году жизни.

Жизнь Андрея Александровича была наполнена интереснейшими делами и событиями. Он участвовал в работе над проектами Отдела программирования ВЦ СОРАН СССР: совместно с Андреем Петровичем Ершовым работал над построением разметки для Операторных алгоритмов, был координатором 1-й фазы проекта БЕТА и вместе с А.Ф. Раром отвечал за Алгол-68, разработал систему электронной подготовки изданий РУБИН. Разработанные им принципы и механизмы: схемы потока данных, электронная подготовка изданий, представление и преобразования шрифтов на ЭВМ, замкнутые открытые операционные обстановки высокого уровня, принцип равнозначности единичного исполнения программного фрагмента и конкретной деятельности, принцип информационной замкнутости.

Работал в проектах: БЕТА, ТАУ, САПФИР, РУБИН, МРАМОР.

Талант Андрея Александровича ярко раскрылся в проекте РУБИН – это система компьютеризации производственных процессов редакции и издательства газеты «Правда». Андрей Александрович в течение нескольких лет изучал производственные процессы издательства, создал лабораторный инструмен-

тальный комплекс, сформировал сильный коллектив, в основном состоящий из его учеников. Андрей Александрович Берс был назначен издательством «Правда» главным конструктором системы РУБИН. Его разработки на десятилетия определили техническую политику в области применения ЭВМ в издательстве.

Андрей Александрович Берс много преподавал: в Летней Школе Юных Программистов, в Новосибирской физико-математической школе, на факультете информационных технологий НГУ, регулярно с лекциями приезжал к нам на факультет компьютерных наук ОмГУ.

Работа Андрея Александровича в нашей редколлегии привела к открытию рубрики «Компьютерные науки».

В его лице мы потеряли мудрого наставника, коллегу и просто друга. Мы запоем Андрей Александровича улыбчивым, энергичным, доброжелательным и в тоже время жёстким и принципиальным, когда рассматриваемые вопросы касались науки. Выражаем искренние соболезнования родным, близким и всем, кто знал Андрея Александровича.

Андрей Александрович Берс был похоронен в Новосибирске.

*Редколлегия журнала  
«Математические структуры  
и моделирование»*

**СВЕДЕНИЯ ОБ АВТОРАХ**

**Бречка Денис Михайлович** – к.т.н., ст. преподаватель, кафедра информационной безопасности, ОмГУ им. Ф.М. Достоевского.

**Гринь Анатолий Гаврилович** – д.ф.-м.н., профессор, кафедра кибернетики, ОмГУ им. Ф.М. Достоевского.

**Гусс Святослав Владимирович** – ассистент, кафедра вычислительных систем, ОмГУ им. Ф.М. Достоевского.

**Гуц Александр Константинович** – д.ф.-м.н., профессор, зав. каф. кибернетики, ОмГУ им. Ф.М. Достоевского.

**Коваленко Юлия Викторовна** – Коваленко Юлия Викторовна - аспирант, кафедра прикладной и вычислительной математики, ОмГУ им. Ф.М. Достоевского.

**Коровянский Алексей Андреевич** – инженер-программист, ООО Al Digit, Омск.

**Кошелева Ольга** – к.ф.-м.н, доцент, педагогический факультет, Техасский университет в Эль-Пасо, США.

**Крейнович Владик** – к.ф.-м.н, профессор, факультет компьютерных наук, Техасский университет в Эль-Пасо, США.

**Лавров Дмитрий Николаевич** – к.т.н, доцент, зав. каф. компьютерных технологий и сетей, ОмГУ им. Ф.М. Достоевского.

**Прохоров Роман Сергеевич** – аспирант, кафедра информационной безопасности, ОмГУ им. Ф.М. Достоевского.

**Пугин Константин Витальевич** – студент, факультет компьютерных наук, ОмГУ им. Ф.М. Достоевского.

**Романов Алексей Николаевич** – к.ф.-м.н, доцент, каф. кибернетики, ОмГУ им. Ф.М. Достоевского.

**Стопкин Сергей Викторович** – студент, факультет компьютерных наук, ОмГУ им. Ф.М. Достоевского.

**Тюменцев Евгений Александрович** – директор ООО «Hello World! Technologies», Омск.

**Федоренко Андрей Александрович** – к.ф.-м.н., научный сотрудник (CR1), Французский национальный центр научных исследований (CNRS), лаборатория физики, Лион, Франция.

**AUTHORS**

**Denis M. Brechka** – Ph.D.(Eng.), Senior Teacher, Chair of Information Security, Omsk State University n.a. F.M. Dostoevskiy.

**Andrey A. Fedorenko** – Ph.D.(Math.), Researcher (CR1) at the French National Centre of Scientific Research (CNRS) Laboratoire de Physique de l'ENS de Lyon, 46, Allée d'Italie, 69007 Lyon, France.

**Anatoly G. Grin** – Doctor of Mathematics, Professor, Cybernetics Chair, Omsk State University n.a. F.M. Dostoevskiy.

**Svyatoslav V. Guss** – Assistant, Chair of Computational System, Omsk State University n.a. F.M. Dostoevskiy.

**Alexander K. Guts** – Doctor of Mathematics, Professor, Head of Cybernetics Chair, Omsk State University n.a. F.M. Dostoevskiy.

**Alexey A. Korovyanskiy** – Engeneer-Programmer, Al Digit Ltd., Omsk.

**Olga Kosheleva** – Ph.D.(Math.), Associate Professor, University of Texas at El Paso, El Paso, TX 79968, USA.

**Uliya V. Kovalenko** – Post-Graduate Student, Chair of Applied and Computational Mathematics, Omsk State University n.a. F.M. Dostoevskiy.

**Vladik Kreinovich** – Ph.D.(Math.), Professor, Computer Science Department, University of Texas at El Paso, El Paso, TX 79968, USA.

**Dmitry N. Lavrov** – Ph.D.(Eng.) Associate Professor, Head of Chair of Computer Technology and Networks, Omsk State University n.a. F.M. Dostoevskiy.

**Roman S. Prohorov** – Post-Graduate Student, Information Security Chair, Omsk State University n.a. F.M. Dostoevskiy.

**Konstantin V. Pugin** – Student, Faculty of Computer Sciences, Omsk State University n.a. F.M. Dostoevskiy.

**Alexy N. Romanov** – Ph.D.(Math.), Associate Professor, Cybernetics Chair, Omsk State University n.a. F.M. Dostoevskiy.

**Sergey V. Stopkin** – Student, Faculty of Computer Sciences, Omsk State University n.a. F.M. Dostoevskiy.

**Eugeniy A. Tyumentcev** – Head of LLC “Hello World! Technologies”, Omsk.

## АННОТАЦИИ

**А.Г. Гринь.** О неравенстве Магды Пелиград.

**Аннотация.** Предлагается новое доказательство вариантов известного неравенства М. Пелиград, в том числе, использующих условия слабой зависимости, отличные от  $\varphi$ -перемешивания.

**Ключевые слова:** Неравенство Пелиград, условия  $\varphi$ -перемешивания,  $\rho$ -перемешивания,  $\lambda$ -перемешивания.

**А.Н. Романов.** Различающие и глобально гиперболические пространства.

**Аннотация.** В работе изучаются условия, которые приводят хронологическое качество пространства-времени к достаточно сильному условию — глобальной гиперболичности. Ключевым моментом является поведение лоренцевой функции расстояния.

**Ключевые слова:** пространство-время, лоренцева функция расстояния.

**А.А. Федоренко.** Поверхностное критическое поведение систем с непрерывной симметрией в присутствии случайных полей.

**Аннотация.** Изучается скейлинговое поведение вблизи поверхности  $d$ -мерной системы с непрерывной  $O(N)$  симметрией в присутствии случайных полей. Выше нижней критической размерности  $d_{lc} = 4$  система демонстрирует фазовый переход из парамагнитного состояния в ферромагнитное, если  $N > N_c = 2.835$ . Ниже нижней критической размерности система находится в квазиупорядоченном состоянии, если  $N < N_c$ , т.е. в состоянии с нулевым параметром порядка, но медленным степенным законом убывания корреляции флуктуаций. Функциональная ренорм-группа позволяет изучить скейлинговое поведение вблизи обыкновенного поверхностного фазового перехода для  $d > d_{lc}$  и поведение корреляции флуктуаций вблизи поверхности в квазиупорядоченной фазе для  $d < d_{lc}$ .

**Ключевые слова:** функциональная ренорм-группа, неупорядоченные системы, случайные поля, поверхностные критические явления.

**О. Кошелева, В. Крейнович.** Скалярно-тензорная теория гравитации Бранса-Дикке может объяснить временную асимметрию физических процессов.

**Аннотация.** Большинство фундаментальных физических уравнений остаются справедливыми, если мы обратим направление времени. Таким образом, если мы начнём с физического процесса (который удовлетворяет этим уравнениям) и обратим направление времени, полученный процесс также удовлетворяет всем уравнениям и, таким образом, также должен быть физически разумным. На практике, однако, многие физические процессы необратимы: например, чашка может развалиться на части, но части не могут волшебным образом собраться и стать целой чашкой. В этой статье мы покажем, что скалярно-тензорная теория гравитации Бранса-Дикке, одно из наиболее широко используемых обобщений общей теории относительности Эйнштейна, является, по сути, асимметричным по времени. И эта асимметрия может объяснить наблюдаемую асимметрию времени физических явлений.

**Ключевые слова:** теория гравитации Бранса-Дикке, асимметрия времени.

**О. Кошелева, В. Крейнович.** Для описания неопределённости эллипсоиды лучше многогранников и, вероятно, лучше прямоугольных параллелепипедов: замечание.

**Аннотация.** Для скаляра множество всех возможных значений является обычно интервалом. Интервал легко представить в памяти компьютера: например, мы можем хранить его двумя конечными точками. Для векторных величин множество возмож-

ных значений может иметь произвольную форму. Точное описание этой формы требует бесконечно много параметров, поэтому в компьютере мы должны использовать семейства множеств конечно-параметрических аппроксимаций. Одним из широко используемых методов для выбора такого семейства является выбор симметричного выпуклого множества и использование его образов при всех линейных преобразованиях. Если взять единичный шар, то мы в конечном итоге будем работать с эллипсоидом, если мы выбираем единичный куб, мы в конечном итоге будем работать с прямоугольными параллелепипедами; мы можем работать и с многогранниками. В этой статье мы покажем, что эллипсоиды приводят к улучшению приближения реальных множеств по сравнению с многогранниками, мы также покажем, что, при разумном допущении, эллипсоиды лучше, чем аппроксимирующие параллелограммы.

**Ключевые слова:** неопределенность, эллипсоиды, интервалы, многогранники.

**А.К. Гуц, Д.Н. Лавров.** Описание DDoS-атаки с помощью катастрофы «сборка».

**Аннотация.** Показано, что DDoS-атаки на компьютерные системы можно описать с помощью катастрофы «сборка».

**Ключевые слова:** DDoS-атака, теория катастроф, сборка.

**Ю.В. Коваленко.** Модель с непрерывным представлением времени для задачи составления расписаний с группировкой машин по технологиям.

**Аннотация.** Рассматривается задача составления расписаний многопродуктового производства. Особенностью постановки является то, что каждый продукт имеет несколько технологий производства, при выполнении которых используется сразу несколько машин, работающих одновременно. Если машина переключается с одной технологии на другую, то необходимо выполнять переналадку. Построены модели частично целочисленного линейного программирования для задачи в общей постановке и для случая, когда длительности переналадки удовлетворяют неравенству треугольника. Для сравнения предложенных моделей проведены численные эксперименты на построенных случайным образом тестовых примерах.

**Ключевые слова:** расписание, переналадки, технологии, целочисленное линейное программирование.

**С.В. Гусс.** Концепция проекта расширяемого обучающего программного средства.

**Аннотация.** В работе даётся описание концепции проекта расширяемого сетевого обучающего программного средства. Предлагаются параметры для управления обучающей системой. Представлены общие элементы.

**Ключевые слова:** обучающее программное средство, элементы повторного использования, параметры обучающих систем.

**С.В. Стопкин, К.В. Пугин, Д.М. Бречка.** Установка программного обеспечения на вычислительный модуль с архитектурой процессора Alpha.

**Аннотация.** Данная работа посвящена выбору и установке программного обеспечения на вычислительный модуль с архитектурой процессора Alpha. Вычислительные модули с такими процессорами входят в состав кластера МВС-1000/128, находящегося в распоряжении факультета компьютерных наук ФГБОУ ВПО «ОмГУ им. Ф.М. Достоевского». В результате работы на вычислительный модуль была установлена операционная система Debian Linux, а также средство поддержки исполнения параллельных программ — OpenMPI.

**Ключевые слова:** вычислительный кластер, операционные системы, аппаратура, па-

раллельные вычисления, MPI, Linux, SRM, Alpha.

**Е.А. Тюменцев.** Автоматизированное тестирование сложности алгоритмов с помощью Mock-объектов.

**Аннотация.** В статье описывается способ тестирования алгоритмической сложности алгоритмов на основе Mock-объектов, позволяющий автоматизировать проверку, исследовать и классифицировать алгоритмы с точки зрения сложности, даже если их реализация неизвестна.

**Ключевые слова:** автоматическое тестирование, алгоритмическая сложность, Mock-объект.

**А.А. Коровянский.** Обзор решений для реализации прототипа замка мобильной биометрической системы доступа M-Block.

**Аннотация.** В статье рассматриваются существующие средства для реализации прототипа замка мобильной биометрической системы доступа M-Block. В результате сравнительного анализа определяется наиболее подходящее решение.

**Ключевые слова:** биометрия, мобильные устройства, биометрическая система доступа, инновационность, ОмГУ, прототип, M-Block, Raspberry Pi, Android, Arduino, AllWinner A1X.

**Р.С. Прохоров.** Бихевиористическая идентификация процессов.

**Аннотация.** В данной статье рассмотрен подход, позволяющий осуществить идентификацию процесса по его поведению. Подход основан на преобразовании потока событий в поток контейнеров событий и использовании такого потока для подачи на вход интеллектуального анализатора, построенного с помощью мультинейронной сети. В качестве «ядра» анализатора используется принципиально новая структура композитной нейронной сети, названная мультинейронной сетью. Подробно описаны варианты используемых преобразований, а также сами используемые преобразования.

**Ключевые слова:** Идентификация процессов, защита информации, бихевиористический анализ, мультинейронная сеть.

## ABSTRACTS

**A.G. Grin.** About inequality of Magda Peligrad.

**Abstract.** We offer a new proof of variants of the inequality of M. Peligrad, including using conditions of weak dependence different, than the  $\varphi$ -mixing.

**Keywords:** inequality of M. Peligrad, conditions of  $\varphi$ -mixing,  $\rho$ -mixing,  $\lambda$ -mixing.

**A.N. Romanov.** Distinguishing and Globally Hyperbolic Space.

**Abstract.** In this paper we study the conditions that lead chronological quality of space-time to a rather stronger condition — global hyperbolicity. The key point is the behavior of Lorentz distance function.

**Keywords:** space-time, Lorentz distance function.

**A.A. Fedorenko.** Surface criticality in random field systems with continuous symmetry.

**Abstract.** We study the surface scaling behavior of a  $d$ -dimensional random field system with continuous  $O(N)$  symmetry. The system undergoes a paramagnetic-ferromagnetic transition above the lower critical dimension  $d_{lc} = 4$  for  $N > N_c = 2.835$ . Below the lower critical dimension and for  $N < N_c$  the system exhibits a quasi-long-range order with zero

order parameter and a power-law decay of correlations. Using functional renormalization group we obtain the surface scaling laws describing the ordinary surface transition for  $d > d_{lc}$  and the behavior of correlations near the surface in the quasi-long-range ordered phase for  $d < d_{lc}$

**Keywords:** functional renormalization group, disordered systems, random fields, surface criticality.

**O. Kosheleva, V. Kreinovich.** Brans-Dicke Scalar-Tensor Theory of Gravitation May Explain Time Asymmetry of Physical Processes.

**Abstract.** Most fundamental physical equations remain valid if we reversed the time order. Thus, if we start with a physical process (which satisfies these equations) and reverse time order, the resulting process also satisfies all the equations and thus, should also be physically reasonable. In practice, however, many physical processes are not reversible: e.g., a cup can break into pieces, but the pieces cannot magically get together and become a whole cup. In this paper, we show that the Brans-Dicke Scalar-Tensor Theory of Gravitation, one of the most widely used generalizations of Einstein's General relativity, is, in effect, time-asymmetric. This time-asymmetry may explain the observed time asymmetry of physical phenomena.

**Keywords:** Brans-Dicke theory of gravitation, time asymmetry.

**O. Kosheleva, V. Kreinovich.** For Describing Uncertainty, Ellipsoids Are Better than Generic Polyhedra and Probably Better than Boxes: A Remark.

**Abstract.** For a single quantity, the set of all possible values is usually an interval. An interval is easy to represent in a computer: e.g., we can store its two endpoints. For several quantities, the set of possible values may have an arbitrary shape. An exact description of this shape requires infinitely many parameters, so in a computer, we have to use a finite-parametric approximation family of sets. One of the widely used methods for selecting such a family is to pick a symmetric convex set and to use its images under all linear transformations. If we pick a unit ball, we end up with ellipsoids; if we pick a unit cube, we end up with boxes and parallelepipeds; we can also pick a polyhedron. In this paper, we show that ellipsoids lead to better approximations of actual sets than generic polyhedra; we also show that, under a reasonable conjecture, ellipsoids are better approximators than boxes.

**Keywords:** uncertainty, ellipsoids, intervals, polyhedra.

**A.K. Guts, D.N. Lavrov.** Description of DDoS-attacks by the cusp catastrophe.

**Abstract.** It is shown that the DDoS-attacks on computer systems can be described by the cusp catastrophe.

**Keywords:** DDoS-attack, catastrophe theory, cusp.

**J.V. Kovalenko.** Continuous-time model for Scheduling with Technology Based Machines Grouping.

**Abstract.** A problem of multi-product scheduling is considered. Each product can be produced by a family of alternative multi-machine technologies. Multi-machine technologies require more than one machine at the same time. A sequence dependent setup time is needed between different technologies. We formulate a mixed integer linear programming model for the general case of the problem and for the case, when the setup times satisfy the triangle inequality. The proposed models are experimentally tested on randomly generated instances.

**Keywords:** scheduling, setup times, technologies, integer linear programming.

**S.V. Guss.** Extensible game-based educational software project.

**Abstract.** The paper is about the extensible game-based educational software project: conception, components, common settings.

**Keywords:** educational software, software reuse, educational software settings.

**S.V. Stopkin, C.V. Pugin, D.M. Brechka.** Installation of the software on computer module with Alpha processor architecture.

**Abstract.** The work is dedicated to selecting and installing the software on the computer module with the Alpha processor architecture. Computing modules with processors of this type are included in the cluster MVC-1000/128, which is at the disposal of Faculty of Computer Science of Omsk State University n.a. F.M. Dostoevskiy. As a result of the work there was installed Debian Linux operating system and OpenMPI — an environment for supporting the parallel programs.

**Keywords:** computer cluster, operating systems, hardware, parallel computing, MPI, Linux, SRM, Alpha.

**E.A. Tyumentsev.** Automated testing of algorithms using Mock objects.

**Abstract.** his article reviews the method of testing the algorithmic complexity of the algorithms based on the Mock objects to automate the test, investigate and classify the complexity of the algorithms, even if their implementation is unknown.

**Keywords:** automated testing, algorithmic complexity, Mock object.

**A.A. Korovyansky.** Overview of solutions for the implementation of the prototype mobile lock biometric access system M-Block.

**Abstract.** The paper reviews existing funds for the implementation of the prototype mobile lock biometric access system M-Block. A comparative analysis determines the most suitable solution.

**Keywords:** biometrics, mobile devices, biometric access control system, innovation, Omsk State University, prototype, M-Block, Raspberry Pi, Android, Arduino, AllWinner A1X.

**R.S. Prokhorov.** Behavioristic process identification.

**Abstract.** In this paper, the approach for the identification of the process by its behavior is considered. The approach is based on the transformation of the events flow to the container flow of events and the use of this flow for inputting to intelligent analyzer built with multineural network. A radically new composite structure of the neural network called multineural network is used as the "core" of the analyzer.

**Keywords:** Process identification, information security, behavioristic analysis, multineural network.

Научный журнал

# Математические структуры И моделирование

Том 27, № 1

Главный редактор

**Д.Н. Лавров**

Корректор

**И.Н. Баловнева**

Художественное оформление

**Д.Н. Лавров**

---

**Адрес научной редакции**

Россия, 644053, Омск-53, ул. Грозненская, 11

Омский государственный университет  
факультет компьютерных наук

E-mail: [lavrov@omsu.ru](mailto:lavrov@omsu.ru)

Электронная версия журнала:

<http://msm.univer.omsk.su>

<http://msm.omsu.ru>



---

Подписано в печать 11.07.13. Формат 60 × 84 1/8.

Усл. печ. л. 14,2. Тираж 125 экз. Заказ № 160.

---

Отпечатано на полиграфической базе издательства ОмГУ им. Ф.М. Достоевского  
644077, г. Омск, пр. Мира, 55А

ISSN 2222-8772



9 772222 877005



13027 >