

# Научно-технический журнал

6 номеров в год

## УЧРЕДИТЕЛЬ

Институт проблем управления  
им. В.А. Трапезникова РАН

## Главный редактор

И.В. Прангишвили

## Заместители главного редактора

А.Н. Шубин, Ф.Ф. Пашенко

## Ответственный секретарь

Л.П. Боровских

## Выпускающий редактор

Л.В. Петракова

## Региональные редсоветы (руководители)

<b>Владивосток</b>	— О.В. Абрамов (4232) 31-02-02
<b>Воронеж</b>	— С.А. Баркалов (0732) 76-40-07
<b>Липецк</b>	— Л.А. Кузнецов (0742) 32-80-44
<b>Минск</b>	— А.В. Тузиков (37517) 284-21-40
<b>Тирасполь</b>	— С.И. Берилл (10-373553) 9-44-87
<b>Уфа</b>	— Б.Г. Ильясов (3472) 23-78-35

## Издатель

ООО «СенСиДат-Контрол»

Ген. директор Н.Н. Кузнецова

Адрес редакции  
117997, ГСП-7, Москва,  
ул. Профсоюзная, д. 65, к. 104.  
Тел./факс (095) 330-42-66,  
тел. (095) 334-92-00

E-mail: [datchik@ipu.ru](mailto:datchik@ipu.ru)  
[www.ipu.ru/period/pu](http://www.ipu.ru/period/pu)

Оригинал-макет  
и электронная версия  
подготовлены  
ООО «ЭЛЕКТРОНИНФОРМ»

Отпечатано с готовых диапозитивов  
в типографии ГКС

Подписано в печать  
28.06.2005 г.

Заказ № РВ405

Журнал зарегистрирован  
в Министерстве  
Российской Федерации  
по делам печати,  
телерадиовещания  
и средств массовых  
коммуникаций

Свидетельство о регистрации  
ПИ №77-11963  
от 06 марта 2002 г.

Журнал входит в Перечень ВАК  
(Бюл. ВАК. — 2004. — № 3)

Подписные индексы:  
81708 в каталоге Роспечати  
38006 в объединенном каталоге  
«Пресса России»

© СенСиДат-Контрол, 2005 г.

# ПРОБЛЕМЫ УПРАВЛЕНИЯ

## 4.2005

## СОДЕРЖАНИЕ

### Информационные технологии в управлении

- Выхованец В. С., Иосенкин В. Я.** Понятийный анализ и контекстная технология программирования . . . . . 2
- Затуливетер Ю. С., Топорищев А. В.** Язык ПАРСЕК: программирование глобально распределенных вычислений в модели исчисления древовидных структур . . . . . 12
- Полетыкин А. Г.** Особенности разработки программного обеспечения сложных интегрированных АСУТП . . . . . 21

### Математические проблемы управления

- Левин В. И.** Логическое моделирование разрывных функций . . . . . 25
- Исмаилов И. Г.** Итерационный алгоритм построения циклов нелинейных автономных систем. Ч. 2. Оценки параметров . . . . . 30

### Надежность и техническая диагностика

- Ведешенков В. А.** Способ самодиагностирования неоднородных цифровых систем . . . . . 33

### Управление в медико-биологических системах

- Червоненкис А. Я.** Применение методов распознавания образов в задачах молекулярной биологии . . . . . 41
- Бабушкина Н. А., Островская Л. А., Рыкова В. А. и др.** Моделирование эффективности действия противоопухолевых препаратов в сверхмалых дозах для оптимизации режимов их введения . . . . . 47

### Управление в социально-экономических системах

- Копнин М. Ю., Кульба В. В., Микрин Е. А.** Структурно-технологический резерв и его использование для повышения устойчивости производственных систем . . . . . 55
- Колемаев В. А., Бережной А. Е.** Моделирование смены технологического уклада . . . . . 61
- Лузгина О. А.** Управление диверсификацией деятельности промышленных предприятий на региональном рынке . . . . . 67
- Макаров Е. И., Головинский П. А.** Управление выбором поставщика в условиях вероятностного прогноза . . . . . 70

### Управление подвижными объектами

- Абрамянц Т. Г., Маслов Е. П., Рубинович Е. Я.** Управление подвижными объектами в условиях искусственно организованной неполноты информации . . . . . 75

### Хроника

- Александр Михайлович Петровский** (к 80-летию со дня рождения) . . . . . 82
- Научные чтения памяти А. М. Петровского** . . . . . 83

## ПОНЯТИЙНЫЙ АНАЛИЗ И КОНТЕКСТНАЯ ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

В. С. Выхованец<sup>(1)</sup>, В. Я. Иосенкин<sup>(2)</sup>

<sup>(1)</sup> *Институт проблем управления им. В. А. Трапезникова, г. Москва;*

<sup>(2)</sup> *Приднестровский государственный университет им. Т. Г. Шевченко, г. Тирасполь*

Рассмотрена контекстная технология программирования, основанная на создании специализированного языка для решения заданного класса прикладных задач путем понятийного анализа предметной области и отражения ее понятийной структуры в понятиях языка. Описаны средства для задания семантики определяемого языка. На примерах программ показана возможность реализации в рамках контекстной технологии некоторых известных методов описания семантики.

### ВВЕДЕНИЕ

Современные информационные технологии характеризуются возрастающей сложностью *информационных систем*, создаваемых в различных областях и предназначенных для хранения, обработки, поиска, распространения, передачи и предоставления информации. Накопленный опыт их проектирования показывает сложность и трудоемкость этого процесса, длительного по времени и требующего высокой квалификации участвующих в нем специалистов.

До сих пор проектирование информационных систем выполняется в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве проектировщиков, их практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках получаемых результатов. Кроме того, в процессе жизненного цикла информационная система подлежит постоянному изменению в соответствии с изменяющимися потребностями пользователей и развитием их представлений о предметной области, что еще более усложняет разработку и сопровождение таких систем.

Структурная методология предоставила в распоряжение разработчиков методы структурной декомпозиции предметной области. Наглядность и строгость *структурного анализа* позволяет разработчикам и пользователям системы формализовать представления, необходимые для адекватной реализации информационной системы.

В структурном анализе и проектировании применяются различные модели, описывающие функциональную структуру системы (модель SADT) [1], последовательность выполняемых действий (модель IDEF3) [2], передачу данных между функциональными процессами (диаграммы потоков данных DFD) [3], отношения между данными (модель “сущность—связь” или ERM) [4].

Однако широкое применение структурного анализа и следование его рекомендациям связано с трудностями создания детальных формальных спецификаций информационной системы, проверки их на полноту и непротиворечивость. Еще большие трудности ожидают разработчиков при попытке изменения этих спецификаций.

На смену структурному анализу и как естественное его развитие пришла методология *объектно ориентированного анализа* и одноименная технология программирования. Объектно ориентированный анализ необходим для создания объектной модели предметной области. Основные принципы объектно ориентированной технологии (децентрализация, контракты, самодостаточность, классификация, бесшовность) [5] реализуются методами наследования, инкапсуляции и полиморфизма при структурной декомпозиции предметной области [6]. При этом используются такие понятия как объект, класс, атрибут, метод, интерфейс и др.

Формальная спецификация результатов объектно ориентированного анализа осуществляется на языке моделирования UML (Unified Modeling Language) [7], который содержит графические и языковые средства для определения, представления, проектирования и доку-



ментирования результатов анализа предметной области. Язык поддерживает стандартный набор *декомпозиционных схем*, реализуемых в виде всевозможных диаграмм и нотаций. В UML отражена как методология структурного анализа в виде диаграмм классов, компонентов и их размещения, так и объектно ориентированная методология, представленная в виде различных моделей поведения объектов: варианты применения, взаимодействие, состояния, деятельность.

Отличительное свойство UML от других средств моделирования (IDEF3, DFD, ERM) — возможность расширения языка, основанная на таких сущностях, как стереотипы, теги и ограничения. Однако главное достоинство объектно ориентированного подхода состоит в том, что создаваемые объектно ориентированные модели более открыты и легче поддаются внесению изменений, поскольку их конструкция базируется на устойчивых формах отражения результатов декомпозиции. Это дает возможность модели развиваться постепенно и не приводит к полной ее переработке даже в случае существенных изменений представлений о предметной области и требований к проектируемой информационной системе.

Известный недостаток современных технологий программирования заключается в наличии достаточно большого семантического (смыслового) разрыва между содержательными представлениями о предметной области и решаемыми задачами и теми средствами, которые заложены в языке программирования для решения задач. Иными словами, *семантический разрыв* — это явление несоответствия решаемых задач тем средствам, которые используются для их решения.

Одна из особенностей семантического разрыва определяется различием системы понятий языка программирования и системы понятий, применяемых для постановки и решения задач в той или иной предметной области. По своей сути любой универсальный язык программирования навязывает разработчику программ некоторую систему понятий, в то время как высокоуровневая модель предметной области с этими понятиями согласуется плохо или не согласуется вообще.

Высокая стоимость информационных систем, их низкая эффективность и надежность, объективная трудоемкость, интеллектуальная и технологическая сложность процесса программирования в некоторой степени являются последствиями семантического разрыва. Для сокращения семантического разрыва используется повышение уровня абстракций языков программирования [8]. Однако последнее снимает только часть проблем, не затрагивая существенным образом понятийную систему языка программирования.

Настоящая статья посвящена контекстной технологии программирования, которая зародилась как объединение объектно ориентированной и фортоподобной технологий, осуществленное путем контекстной интерпретации слов фортоподобного языка [9] и на основе понятийного анализа предметной области.

С целью сокращения семантического разрыва средствами контекстной технологии создается специализированный язык, отражающий понятийную структуру предметной области и класс решаемых задач. Такой подход основывается на допущении, что уже в процессе

изучения предметной области и специфики заданного класса прикладных задач, еще до начала программирования, формируется система понятий, приспособленная для постановки и решения задач, которая и найдет свое отражение в создаваемом языке.

Другой базовый принцип, именем которого названа представляемая технология, состоит в контекстной интерпретации лексем. Контекстная интерпретация как принцип заключается в определении семантического (смыслового) значения лексем специализированного языка в зависимости от окружающего их контекста. В отличие от генераторов компиляторов [10], у которых возможности задания контекстных условий, необходимых для реализации языков программирования высокого уровня, достаточно бедны, в контекстной технологии контекстные условия не только систематически задаются, но и естественным образом используются. Это позволяет не только улучшить выразительные возможности определяемого языка, но и повысить его уровень.

И, наконец, в связи с зарождением контекстной технологии как некоторого развития объектно ориентированной, ей не чужды все те принципы и методы, на которых основана последняя.

## 1. КОНТЕКСТНАЯ ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Данную технологию будем рассматривать как метаязык и систему программирования, которые объединены в рамках единой методологии, определяемой как совокупность методов, применяемых в процессе разработки программ и объединенных одним общим подходом.

### 1.1. Понятийная модель и прикладные задачи

Под *предметной областью* будем понимать выделенный фрагмент реальной действительности, представляемый в нашем сознании совокупностью некоторых сущностей (предметов, процессов, явлений, задач и т. д.), которые могут быть уникально идентифицированы и описаны.

Наиболее естественным способом описания сущностей предметной области видится соотнесение с ними совокупности понятий, образующих ее понятийную структуру [11]. В процессе программирования будем строить ее *понятийную модель*, для чего предусмотрим специфические языковые средства.

Понятийная модель строится путём выявления и выражения *понятийной структуры* предметной области в объеме, достаточном для решения некоторого класса прикладных задач. В отличие от других технологий, например структурной, основанной на структурном анализе предметной области, и объектно ориентированной, основанной на объектном анализе, контекстную технологию программирования определим на основе понятийного анализа предметной области.

В качестве программы выступает понятийная модель предметной области, дополненная описанием решения прикладной задачи. При этом одна и та же модель может применяться для решения разных или нескольких прикладных задач. В понятийной модели условно выделяются четыре части:

- понятийная структура предметной области;

- синтаксические правила выражения понятий в тексте;
- описание семантики правил выражения понятий;
- описание процесса компиляции.

Понятийная структура и синтаксические правила описываются на декларируемом в контекстной технологии *метаязыке*, а описание семантики и решаемой задачи выполняются на специализированном языке, задаваемом понятийной структурой и синтаксическими правилами. Тогда в результате описания понятийной структуры предметной области и синтаксических правил для выражения понятий определяется некоторый специализированный язык программирования, на котором осуществляется как решение прикладной задачи, так и частичное описание его семантики.

Таким образом, понятийная структура и синтаксические правила задают синтаксис определяемого в модели специализированного языка. *Определяемый язык*, в свою очередь, применяется как для задания семантики синтаксических правил, так и для описания решения прикладной задачи.

## 1.2. Грамматический разбор и компиляция

Компиляция понятийной структуры и синтаксических правил в предлагаемой технологии осуществляется традиционными методами на основе определения синтаксиса некоторого метаязыка в виде формальной грамматики и содержательного описания его семантики. Компиляция описаний семантики для правил выражения понятий и текста решения прикладной задачи осуществляется создаваемым в процессе программирования специализированным компилятором. Последнее обеспечивается тем, что любое синтаксическое правило может дополняться описанием процесса его компиляции, которое также выражается на определяемом языке.

Синтаксический анализ и грамматический разбор части программы, написанной на метаязыке контекстной технологии, реализуется известными средствами.

Синтаксический анализ и грамматический разбор описаний семантики правил, решаемой задачи и процесса компиляции выполняется специально разработанными для контекстной технологии методами синтаксического анализа и грамматического разбора.

Для повышения эффективности компилятора системы контекстного программирования применяется *разнесенный грамматический разбор*. Последний основан на контекстной интерпретации лексем и заключается в разделении определения применимости синтаксических правил на две части: на контекстное сопоставление, осуществляемое при просмотре назад, и структурное сопоставление, выполняемое при просмотре вперед. Описание компилятора выходит за рамки настоящей статьи, в связи с чем процесс компиляции здесь не рассматривается.

## 1.3. Семантический разрыв

Как следствие формирования в процессе программирования специализированного языка для описания решаемой задачи, применение контекстной технологии программирования направлено на сокращение семантического разрыва между высокоуровневой моделью предметной областью и языком программирования. Более того, в результате приближения выразительных средств

языка программирования к постановке и решению прикладных задач следует ожидать повышения качества и надежности информационных систем, созданных на основе контекстной технологии.

Другой особенностью контекстной технологии, вытекающей из наличия выразительных средств описания процесса компиляции, следует назвать возможность использования системы контекстного программирования совместно с любой из известных целевых платформ, например:

- с микроконтроллерами, не имеющими операционной системы;
- с вычислительными системами, имеющими развитую операционную среду;
- с платформенно-независимыми виртуальными машинами;
- с другими системами программирования.

Очевидно, в случае аппаратной платформы привязка определяемого в модели специализированного языка к целевой платформе осуществляется низкоуровневыми средствами, например, путем прямой генерации кода. Другой крайний случай заключается в привязке определяемого языка к целевой платформе с помощью другой системы программирования, и тогда определяемый язык описывается на языке, поддерживаемом этой системой. Для исследования контекстной технологии авторами реализована система программирования, поддерживающая эти крайние случаи.

Не претендуя на полноту изложения, кратко опишем только те процессы, которые актуальны в рамках настоящей статьи и существенным образом отличают контекстную технологию программирования от других технологий. Для раскрытия содержания контекстной технологии рассмотрим более детально составные части понятийной модели.

## 2. МЕТАЯЗЫК КОНТЕКСТНОЙ ТЕХНОЛОГИИ

### 2.1. Базовые понятия

*Терм* определим как элементарную синтаксическую единицу текста на некотором формальном языке, которая выражается последовательностью *знаков алфавита* этого языка. Сам алфавит и входящие в него знаки назовем *терминальными*.

*Текстом* будем называть последовательность лексем, предназначенную для выражения некоторого сложного смысла. *Лексему* определим как элементарную смысловую (семантическую) единицу текста, выраженную одним или несколькими терминами. Иными словами, лексема — это множество термов вместе с приписанным (сопоставленным) им смыслом. Лексема, выраженная одним терминальным знаком, называется *символом*. Текст, в отличие от лексем, предполагает свое деление на смысловые части, в то время как лексемы такого деления не допускают.

Для выражения синтаксиса текстов на определяемом языке воспользуемся известным формализмом контекстно-свободных грамматик [12]. *Контекстно-свободной грамматикой* называется формальная система  $\langle T, N, S, I \rangle$ , состоящая из терминального алфавита  $T$ , нетерминального алфавита  $N$ , множества правил вывода, или предложений,  $S$  вида  $A \rightarrow \alpha$  и выделенного не-



терминального знака  $l$ , называемого аксиомой, где  $A$  — нетерминальный знак, а  $\alpha$  — конечная последовательность терминальных и нетерминальных знаков.

Для обозначения термов в правилах грамматики будем пользоваться одинарными кавычками. Например, 'x' означает терм, состоящий из одного знака  $x$ . В свою очередь, последовательностями терминальных знаков, не заключенными в одинарные кавычки, обозначим нетерминальные знаки грамматики. Для разделения нетерминальных знаков между собой будем пользоваться пробелами, для чего запретим появление пробела в обозначении нетерминальных знаков.

Как обычно, каждому нетерминальному знаку грамматики сопоставим некоторое понятие, называемое нетерминальным понятием языка, которое для краткости будем именовать просто понятием. Тогда правила вывода или предложения грамматики могут быть интерпретированы как синтаксические правила выражения понятий в тексте.

Под *контекстной интерпретацией* термина будем понимать определение семантического значения термина по его месту в тексте. В общем случае один и тот же терм может быть сопоставлен различным лексемам, т. е. одна и та же последовательность терминальных знаков может служить носителем различных смыслов. Следовательно, при контекстной интерпретации семантическое значение термина в общем случае определяется окружающим его контекстом.

**Пример 1.** Пусть в грамматике языка имеется два правила с термом 'x':  $C \rightarrow A 'x' B$  и  $F \rightarrow 'x' E$ , где  $A, B, C, E$  и  $F$  — некоторые понятия, а знак  $\rightarrow$  разделяет определяемое понятие (слева) от понятий и термов (справа), используемых для его выражения в тексте. Таким образом, понятие  $C$  определяется первым предложением, а понятие  $F$  — вторым. В свою очередь, где-то имеются предложения, определяющие понятия  $A, B$  и  $E$ .

Анализ приведенных предложений показывает, что если терм 'x' встретится в окружении последовательности терминальных знаков, выражающих понятия  $A$  и  $B$ , то он интерпретируется как лексема  $x$  первого предложения. В свою очередь,  $A 'x' B$  (как и порожденная в соответствии с ней последовательность термов) выражает пример понятия  $C$ . Если терм 'x' встретится в контексте 'x' E, то он сопоставляется с лексемой  $x$  второго предложения, а соответствующая часть текста выражает пример понятия  $F$ . Очевидно, если ни один из разрешенных контекстов термина 'x' не распознан, то это говорит об ошибке в тексте. ♦

В демонстрационных целях выберем в качестве языка, определяемого средствами контекстной технологии, язык булевых выражений, описываемый обозримой в рамках настоящей статьи грамматикой.

**Пример 2.** Язык булевых выражений зададим следующей грамматикой:

```
Boolean → 'false'
Boolean → 'true'
Boolean → "[A-Za-z][A-Za-z0-9]*"
Boolean → '(' Boolean ')'
Boolean → 'not' Boolean
Boolean → Boolean 'and' Boolean
Boolean → Boolean 'or' Boolean
Boolean → Boolean 'imp' Boolean,
```

где нетерминальный алфавит состоит из одного понятия Boolean (аксиома грамматики).

Например, булево выражение '(not x or y) and z' порождается этой грамматикой. Заметим, что третье предложение задает

синтаксис выражения переменных, определенный на языке регулярных выражений [13], где регулярное выражение представлено в виде терминального шаблона и для отличия от термина заключено в двойные кавычки. ♦

## 2.2. Грамматика метаязыка

*Метаязык* контекстной технологии определим приведенной на рисунке грамматикой, которая описана с точностью до обозначенных курсивом нетерминальных знаков *notion*, *aspect* и *string*. Описываемый метаязык является развитием контекстного языка из работы [14].

Текст программы *program* состоит из последовательности *деклараций* *declaration* и *ситуаций* *situation* (строка 1). В декларативной части (последовательности деклараций) определяется язык, на котором в ситуационной части описывается решение некоторой прикладной задачи.

Отдельная декларация представляет собой законченный фрагмент определения языка, а ситуация описывает решение прикладной задачи полностью, если в ранее следующих декларациях определяемый язык задан полностью, или частично, когда язык определен в объеме некоторого своего подязыка, уже достаточного для описания части решения.

Декларативная часть состоит из описаний сущностей предметной области *essence* (строка 2). Каждой сущности присваивается имя *notion* нетерминального понятия определяемого языка, а само понятие задается как находящееся в отношении агрегации *aggregation* и обобщения *generalization* с системой *notions* других, ранее определенных понятий (строки 3—5). Для разделения понятий в списке *notions* используется один или несколько пробелов *space* (строка 17), а при их отсутствии — пустой знак *empty*, который определен как отсутствие каких-либо знаков или как последовательность пробелов (строка 18).

Описание любой сущности *essence* одновременно содержит указание как на обобщение, так и на агрегацию (строка 3). Более того, любая сущность может быть определена с помощью этих и только этих абстракций (строка 3, первое правило). Для выражения перекрестных и рекурсивных связей между сущностями используется их предварительное объявление, которое не включает конструкцию *aggregation* (строка 3, второе правило). Таким образом, описание *сущности*, в отличие от описания составляющего его понятия, содержит определение существующего для него отношения обобщения.

Каждое описание *description* содержит предложение *sentence* (строка 7), состоящее из последовательности элементов *item* (строка 8): имен понятий *notion*, лексем *lexeme* и текстов компиляции *compilation* (строка 9). Лексема является терминальным понятием определяемого языка. Для выражения лексем могут использоваться как термы *term*, так и множества термов, задаваемые на языке регулярных выражений [13] в виде терминальных шаблонов *pattern* (строки 10—12).

С каждым предложением связывается определяемое *понятие-результат*, именем которого названа описываемая сущность (строка 3), определяемая предложением полностью, когда больше нет предложений с тем же понятием-результатом, или частично, если такие предложения имеются (строки 6 и 7).



1	program	→	declaration situation   declaration situation program
2	declaration	→	essence   essence declaration
3	essence	→	generalization <i>notion</i> space aggregation   generalization <i>notion</i>
4	generalization	→	'(' notions ')'   '(' empty ')'
5	notions	→	notion   notion space notions
6	aggregation	→	description   description aggregation
7	description	→	sentence semantics
8	sentence	→	item   item space sentence
9	item	→	notion   lexeme   compilation
10	lexeme	→	term   pattern
11	term	→	" string "
12	pattern	→	"" string ""
13	semantics	→	imperative   imperative space semantics
14	imperative	→	'{ text }'   <i>aspect</i> '{ text }'
15	text	→	phrase   phrase text
16	phrase	→	empty   string   <i>aspect</i> '{ text }'
17	space	→	' '   ' ' space
18	empty	→	"   space
19	compilation	→	'[ text ]'
20	situation	→	'[ text ]'

### Порождающая грамматика метаязыка

**Пример 3.** Представим грамматику из примера 2 на метаязыке контекстной технологии в виде декларации сущности Boolean:

```
() Boolean
  'false' {}
  'true' {}
  "[A-Za-z][A-Za-z0-9]*" {}
  '(' Boolean ')' {}
  'not' Boolean {}
  Boolean 'and' Boolean {}
  Boolean 'or' Boolean {}
  Boolean 'imp' Boolean {},
```

где для разделения сущностей, если их несколько, используются пустые круглые скобки конструкции *generalization*, а для разделения предложений внутри определения сущности — пустые фигурные скобки конструкции *semantics*. Для приведенного фрагмента понятийной модели может быть задана ситуация, например, в виде текста '(not x or y) and z', являющегося решением некоторой прикладной задачи, например, вычислением значения булевого выражения. ♦

По своей сути предложение *sentence* является представлением правил грамматики определяемого языка на метаязыке контекстной технологии с тем отличием, что для задания особенностей компиляции предложения используются конструкции *compilation* (строка 19). Их роль — указание компилятору системы контекстного программирования на действия, которые необходимо выполнить в процессе распознавания предложения в тексте, например, проверку более сложных контекстных условий, чем это задается средствами метаязыка (контекстным условиям посвящен п. 2.6). Описание системы контекстного программирования выходит за рамки настоящей статьи, и конструкция *compilation* более детально не рассматривается.

Рассмотрим теперь семантику описания отношений обобщения и агрегации. Оставшиеся нераскрытыми конструкции метаязыка и их семантика будут описаны позже.

### 2.3. Понятийный анализ

Абстрагирование понятий представляет собой основную механизм, с помощью которого человек познает окружающий мир. *Абстракция* — это выделение существенных признаков и связей понятий и игнорирование несущественных. Абстрагирование является основным методическим приемом понятийного анализа и используется при разработке понятийной модели. Абстрагирование позволяет декомпозировать предметную область на значимые сущности и построить ее понятийную структуру, адекватную решаемым задачам.

*Понятийный анализ* определим как методику построения понятийной структуры, основанную на абстрагировании понятий и осуществляемую путем сравнения и сопоставления сущностей предметной области с целью выявления их общих признаков, выделения понятий, объединения понятий на основе их сходства или подобия и т. п.

**Пример 4.** Детализируем описание языка булевых выражений введением дополнительных понятий:

```
() Constant
  'false' {}
  'true' {}
() Variable
  "[A-Za-z][A-Za-z0-9]*" {}
() Logic
  Variable {}
  '(' Boolean ')' {}
```



- () Negation
  - 'not' Constant {}
  - 'not' Logic {}
- () Conjunction
  - Negation 'and' Negation {}
- () Disjunction
  - Conjunction 'or' Conjunction {}
- () Boolean
  - Disjunction 'imp' Disjunction {}

В рассматриваемом примере Constant обозначает понятие “логическая константа”, Variable — понятие “пропозиционная переменная”, а Logic — понятие “логическое значение”. Остальные понятия соответствуют более сложным представлениям, правила выражения которых декларируются в примере: Negation обозначает понятие “отрицание”, Conjunction — понятие “конъюнкция”, а Disjunction — понятие “дизъюнкция”.

Очевидно, наиболее сложным понятием является Boolean, обозначающее понятие “булево выражение”. Понятия Constant, Logic, Negation, Conjunction и Disjunction являются частными случаями Boolean. Однако выражение этих частных случаев осуществляется по-разному, что и задается в виде соответствующих предложений. ♦

Из примера видно, что имеются различные взаимосвязи понятий, появляющиеся в результате их абстрагирования. Известны следующие основные типы абстракции [15]: ассоциация, обобщение, типизация и агрегация, причем фундаментальными являются только обобщение и агрегация. Абстракция типизации является частным случаем обобщения, а абстракция ассоциации эквивалентна абстракции агрегации по способу формирования [16]. Рассмотрим средства выражения обобщения и агрегации в понятийной модели, не останавливаясь на особенностях выражения частных типов абстракции.

#### 2.4. Абстракция обобщения

*Обобщение* понятий — это форма порождения нового понятия на основе одного или нескольких подобных понятий, когда порождаемое понятие сохраняет общие признаки исходных понятий, но игнорирует их более тонкие различия. Для обобщения возможен противоположный процесс, когда исходное понятие делится на несколько более узких. Такой процесс называется *специализацией*, или *ограничением* понятий.

Следовательно, при обобщении подобные видовые понятия соотносятся с родовым, а при специализации, наоборот — родовые понятия делятся на видовые.

**Пример 5.** Выразим абстракцию обобщения в понятийной модели из Примера 4. В итоге получим:

- () Constant
  - 'false' {}
  - 'true' {}
- () Variable
  - "[A-Za-z][A-Za-z0-9]\*" {}
- (Variable) Logic
  - Variable {}
  - (' Boolean ') {}
- (Constant Logic) Negation
  - 'not' Logic {}
  - 'not' Constant {}
- (Negation) Conjunction
  - Negation 'and' Negation {}
- (Conjunction) Disjunction
  - Conjunction 'or' Conjunction {}
- (Disjunction) Boolean
  - Disjunction 'imp' Disjunction {}

Видно, что для описания понятия Negation необходимо понятие Logic и Constant, т. е. понятие Negation получено в результате обобщения понятий Logic и Constant. В свою очередь, понятия Logic и Constant являются конкретизацией понятия Negation. Последнее означает, любое выражение в тексте понятий Logic или Constant является и выражением некоторого частного случая обобщающего понятия Negation. ♦

Метаязык понятийной модели позволяет выразить произвольные обобщения (строки 3 и 4). При этом между понятиями устанавливается связь типа “есть подкласс”. Если понятия видового уровня связаны отношением обобщения только с одним понятием родового уровня, то возникает древовидная структура обобщения. В противном случае имеем некоторую сетевую структуру. Для указания на отсутствие обобщения у некоторого понятия используются пустые круглые скобки (строка 4). В этом случае понятие считается обобщенным от пустого или универсального понятия empty, обозначаемого как '()’.

Обобщение несколько похоже на наследование, используемое в объектно ориентированных языках, но к последнему не сводится. При выявлении родовых понятий обобщение указывает на его видовые составляющие. Это позволяет при задании родовых понятий не повторять описания его видовых реализаций. В объектно ориентированных языках базовый (родовой) класс реализует общие свойства и методы, которые распространяются на все наследуемые (видовые) классы. Однако базовый класс теряет при этом знание о своих видовых реализациях. Для решения этой проблемы в объектно ориентированных языках был введен механизм виртуальных методов.

#### 2.5. Абстракция агрегации

*Агрегация* понятий используется в тех случаях, когда вновь порождаемое сложное понятие включает исходные понятия в качестве своих составных частей. Абстракция агрегации позволяет выразить семантику внутренних связей, существующих между отдельными сущностями предметной области. При этом структура сложного понятия раскрывается путем его расчленения на совокупность составляющих его понятий. Процесс, противоположный агрегации, называется *декомпозицией*.

Агрегация в понятийной модели выражается предложениями sentence, задающие связь агрегируемых понятий с понятием-результатом (строки 7 и 8, см. рисунок). Конструкция aggregation позволяет выразить агрегацию одного понятия или когда в качестве агрегируемых понятий используется одна или несколько лексем. Последнее соответствует агрегации без предварительного объявления агрегируемых понятий путем их прямого (непосредственного) выражения в тексте. Таким образом, лексема может рассматриваться как *терминальное понятие*, имеющее прямое (непосредственное) выражение в тексте программы в виде одной или нескольких терминальных строк.

**Пример 6.** Продемонстрируем агрегацию понятий на примере описания комплексных чисел:

- (Real) Complex
  - (' Real ',' Real ') {}
  - Real '+' Real 'i' {}

где Real — понятие действительного числа. Приведенные предложения определяют два способа выражения комплексных чисел как состоящих из двух действительных, например

'(1, 2)' и '1 + 2i'. Заметим, что понятие Complex обобщено от понятия Real. Это позволяет комплексные числа выражать действительными. Например, терм '2' в соответствующем контексте может быть интерпретирован как комплексное число. ♦

Абстракция агрегации позволяет конструировать новые понятия на основе других, ранее определенных, причем выполнять такое конструирование в каждом предложении по-своему. Последнее отличает агрегацию в контекстной технологии от агрегации в объектно ориентированных языках, где сложный класс конструируется путем простого и единственного перечисления агрегированных в него подклассов.

Таким образом, конструкция essence (строка 3) позволяет рассматривать абстракции агрегации и обобщения в более широком смысле, чем это определено и использовано в объектно ориентированной технологии программирования.

### 2.6. Контекстные условия

Известно, что контекстно-свободные грамматики не отражают синтаксиса современных языков программирования. Это обстоятельство вызвано тем, что эти языки не являются контекстно-свободными и имеют более сложную синтаксическую структуру [17]. Синтаксические правила, которые не описываются контекстно-свободными грамматиками, называются *контекстными условиями*.

Проверку контекстных условий связывают с созданием и использованием терминальных понятий языка, которые имеют различную семантику. Задача контекстного анализа заключается в установлении правильности использования таких понятий, а наиболее часто решаемая задача состоит в определении существования терминального понятия и соответствия его использования одному из возможных для него контекстов.

Контекстные условия в метаязыке задаются для имен понятий notion, имен аспектов aspect и лексем lexeme. На рисунке места определения контекстных условий выделены курсивом.

Определение имени понятия *notion* как терминального понятия метаязыка осуществляется при описании сущности (строка 3). После такого определения имена понятий notion могут появляться при описании обобщения в списке notions (строки 4 и 5) и в качестве item при описании предложений (строки 8 и 9).

Имена аспектов *aspect* служат для задания имен конструкций imperative (строка 14), назначение которых — задание той или иной семантической интерпретации фрагмента текста phrase (строка 16) при описании императивов (строка 14), текста компиляции (строка 19) и ситуаций (строка 20). Более подробно императивы и аспекты будут рассмотрены в § 3.

Лексемы lexeme в метаязыке задаются в виде термов term и шаблонов pattern (строка 10). Как терм, так и шаблон используются для задания терминальных понятий *string* определяемых в модели языка (строки 11 и 12). Причем последние сохраняются не в таблице идентификаторов, как это было для понятий и аспектов, а непосредственно в структуре описываемого предложения sentence (строки 8 и 9).

В заключение заметим, что имена понятий notion являются терминальными понятиями метаязыка и одновременно нетерминальными понятиями определяемого

языка. Имена аспектов *aspect*, как и имена понятий *notion*, являются терминальными понятиями для метаязыка, но в определяемом языке аспекты интерпретируются как терминальные понятия. В свою очередь, лексемы *lexeme* представляются терминальными понятиями *string* определяемого языка.

## 3. СРЕДСТВА ОПИСАНИЯ СЕМАНТИКИ

Под *семантикой* обычно понимается смысловая или содержательная интерпретация текста, в то время как форма представления или структура текста задаются его синтаксисом [18]. В свою очередь, семантика языков определяется совокупностью правил и соглашений, устанавливаемых описанием языка и предназначенных для выявления смысла текстов на этом языке с целью их интерпретации человеком или автоматом [19].

По характеру семантики можно выделить два наклонения: повелительное или *императивное*, представленное операторами (командами, предписаниями), и изъявительное или *декларативное*, представленное только описаниями. В одних языках, называемых императивными, преобладает описание действий, т. е. процесса, позволяющего получить результат (языки FORTRAN, ALGOL, PASCAL, C, ADA и др.). Другие языки, называемые декларативными, предполагают не построение (вычисление) результата, а описание (декларацию) свойств решения; на основе этих данных система программирования сама строит некоторую последовательность действий, необходимую для получения результата (языки PROLOG, LISP, и др.).

### 3.1. Императивы

Семантику предложений понятийной модели будем задавать в виде текста, который построен по правилам определяемого языка и является описанием решения некоторой задачи в аспекте различных проблемных ситуаций. Для этого каждое предложение sentence дополним описанием его семантики semantics, которую выразим как совокупность *императивов imperative* (строки 7 и 13). Императив задает *семантическую интерпретацию* понятия, реализуемую после компиляции предложения в виде некоторой последовательности действий, которую система программирования выполняет всякий раз, когда понятие выражается этим предложением (строка 14).

**Пример 7.** Для модели из Примера 5 зададим императивы, ориентированные на вычисление булевых выражений. Для простоты понимания примера императивы определим в виде последовательности ассемблерных команд достаточно распространенной аппаратной платформы на базе процессора Intel [20].

```
() Variable
    "[A-Za-z][A-Za-z0-9]*" { ... }
() Constant
    'false' { mov eax, 0; push eax }
    'true' { mov eax, -1; push eax }
(Variable) Logic
    Variable { pop ebx; mov eax, [ebx]; push eax }
    Integer { pop eax; cmp eax, 0; je label; mov eax, -1; label:
    push eax }
    (' Boolean ') {}
(Constant Logic) Negation
    'not' Logic { pop eax; not eax; push eax }
```





(Negation) Conjunction  
 Negation 'and' Negation { pop eax; pop edx; and eax, edx;  
 push eax }  
 (Conjunction) Disjunction  
 Conjunction 'or' Conjunction { pop eax; pop edx; or eax, edx;  
 push eax }  
 (Disjunction) Boolean  
 Disjunction 'imp' Disjunction { }

В примере для доступа к данным и организации их временного хранения использован аппаратный стек, а для хранения переменных — память произвольного доступа с линейной организацией. В предложении "[A-Za-z][A-Za-z0-9]\*" текст императива (не показан) описывает создание переменной, путем включения имени переменной в таблицу идентификаторов и выделения для ее хранения памяти требуемого объема.

Логические константы 'false' и 'true' реализованы как занесение нуля и минус единицы на вершину стека, для чего использованы команды загрузки в регистр константы и записи регистра в стек.

Переменная определена адресом ячейки памяти, в которой хранится ее текущее значение. Для получения значения логической переменной Variable ее адрес извлекается из стека, содержимое адресуемой ячейки пересылается в регистр, который затем сохраняется в стеке.

Для преобразования целого числа в булево значение использовано предложение () Logic Integer. Если число на вершине стека не равно нулю, то в регистр записывается минус единица, в противном случае там уже имеется требуемое значение. Результат преобразования сохраняется в стеке.

В свою очередь, предложение (' Boolean ') не нуждается в императиве, так как его роль заключается в задании правил разбора и приоритета булева выражения, заключенного в круглые скобки. При изменении решаемой задачи возможно появление императивов и у этого предложения. Реализация остальных предложений не вызывает труда и понятна из текста примера. ♦

В контекстной технологии императивы могут представляться как в виде последовательности команд целевой платформы, так и программой на некотором целевом языке программирования. Система контекстного программирования выполняет трансляцию текста императива в код целевой платформы непосредственно или организует такую трансляцию путем вызова соответствующих средств целевой системы программирования. Причем в рамках одного императива требуется только один тип вызова: либо вызов ассемблера целевой платформы, который может быть как внешним, так и внутренним по отношению к системе контекстного программирования, либо вызов компилятора целевой системы программирования, который, как правило, является внешним. По своей сути императивы представляют собой единицы вызова целевого языка, в то время как предложение — описание синтаксиса таких вызовов.

### 3.2. Аспекты

Понятно, что понятийная структура предметной области, представленная в декларативном описании, может быть общей для нескольких задач. Однако решение таких задач может преследовать различные цели, в том числе и взаимоисключающие. Для отражения этой ситуации в контекстной технологии предусмотрена возможность задания для одного предложения нескольких императивов imperative (строка 13) Для отличия одного императива от другого они именуются и задаются в виде aspect {...}, где aspect — имя определяемого императива,

который соответствует именованной семантической интерпретации понятия или его аспекту (строка 14).

**Пример 8.** Рассмотрим интерпретацию вычисления булевых выражений, связанную с другим кодированием булевых значений. В Примере 7 логический ноль кодировался арифметическим нулем, а логическая единица минус единицей. Определим новую семантическую интерпретацию модели или ее аспект с именем bit, когда логическая единица кодируется арифметической единицей.

```
() Variable
"[A-Za-z][A-Za-z0-9]*" { ... }
() Constant
'false' { ... }
'true' { ... } bit { mov eax, 1; push eax }
(Variable) Logic
Variable { ... }
Integer { ... } bit { pop eax; and eax, 1; push eax }
(' Boolean ') { }
(Constant Logic) Negation
'not' Constant { ... }
'not' Logic { ... }
(Negation) Conjunction
Negation 'and' Negation { ... }
(Conjunction) Disjunction
Conjunction 'or' Conjunction { ... }
(Disjunction) Boolean
Disjunction 'imp' Disjunction { ... } ♦
```

При использовании именованных императивов, определяемых содержательной постановкой задачи, ситуационное описание необходимо дополнить указанием на одну из возможных его семантических интерпретаций. В итоге, ситуационная часть модели может стать независимой от содержательной интерпретации конкретной прикладной задачи и задавать некоторое общее решение для целого класса задач.

### 3.3. Семантическая замкнутость

Для обеспечения полноты и целостности понятийной модели во всех ее частях (ситуация, семантика, компиляция) целесообразно требовать применения декларируемых языковых конструкций не только для описания ситуационной части (text в строке 20), но и для задания семантики предложений (text в строке 14), а также для описания процесса компиляции (text в строке 19). Тем самым будет реализована семантическая замкнутость описания, что позволит использовать для обработки этих описаний одни и те же средства.

Далее будем рассматривать императивы как текст text, состоящий из фраз phrase (строка 15), построенный по правилам, задаваемым в декларации (строка 16). Очевидно, для привязки модели к целевой платформе необходимо некоторое множество предложений реализовать низкоуровневыми средствами или на целевом языке.

**Пример 9.** Анализ Примера 7 показывает, что все предложения, кроме последнего, реализованы низкоуровневыми средствами. Причем этих предложений уже достаточно для определения семантики последнего предложения средствами самой модели. Для операции импликации  $\rightarrow$  справедливо тождество  $a \rightarrow b = \bar{a} \vee b$ . Отсюда получаем:

```
(Disjunction) Boolean
Disjunction 'imp' Disjunction { not %1 or %2 },
```

где %1 — интерпретируется как операнд, соответствующий первому понятию предложения (первое вхождение Disjunction), %2 — второму. ♦

Из рассмотренного примера следует, что семантически замкнутое императивное описание должно быть

представлено текстом, построенным по правилам, определенным в декларативной части. Более того, семантическая интерпретация этого текста должна соответствовать (прямо или косвенно) понятию-результату. Прямое соответствие означает, что текст императивной части выражает понятие-результат непосредственно, а косвенное — что понятие, выраженное текстом императивного описания, может быть преобразовано в понятие-результат на основе связей обобщения или агрегации.

В свою очередь, текст компиляции (строка 19) и текст ситуационной части (строка 20) никакого понятия не выражают и, как следствие этого, их результат должен соответствовать отсутствующему понятию empty, обозначаемому пустыми круглыми скобками (строка 4).

### Примеры описания семантики

На данный момент не существует универсального, общепринятого метода описания семантики [10]. Перечислим наиболее распространенные методы. Денотационный метод основывается на функциональных вычислениях, в которых встроенные операции языка отображаются в однозначные математические объекты, которые затем применяются для описания семантики языковых конструкций. Аксиоматический метод базируется на исчислении предикатов, где результат вычисления описывается через взаимосвязь переменных до и после применения конструкций языка. И, наконец, в операционном методе операции языка описываются через команды некоторой абстрактной машины.

Известно несколько подходов для задания семантики языков, описываемых контекстно-свободными грамматиками: W-грамматики, венский метаязык, аксиоматический и денотационный методы, синтаксически управляемые схемы, атрибутные транслирующие грамматики [21].

Так как в рамках контекстной технологии заявлен некоторый универсальный метод для описания семантики языков программирования, покажем на примерах реализацию двух наиболее распространенных подходов: синтаксически управляемых схем и атрибутных грамматик.

Синтаксически управляемые схемы [22] и атрибутные грамматики [19] позволяют задавать соответствия между текстами входного и выходного языков, называемые *переводом*. Такие соответствия отражают структурные или синтаксические свойства входных и выходных текстов.

**Пример 10.** Рассмотрим пример реализации в рамках контекстной технологии синтаксически управляемого перевода на примере формального дифференцирования выражений, включающих в себя целочисленные константы, переменную  $x$ , функции  $\sin$  и  $\cos$ , а также алгебраические операции: изменение знака  $-$ , сложение  $+$ , вычитание  $-$  и умножение  $*$ .

Свяжем с каждым предложением модели два перевода, формируемые именованным императивом и именованным императивом  $\text{dif}$ . Неименованный императив указывает на то, что выражение не дифференцируется, а именованный императив  $\text{dif}$  — что выражение необходимо продифференцировать. Формальная производная некоторой строки  $s$  — это  $\text{dif}\{s\}$ , где  $\text{dif}\{\dots\}$  задает имя семантической интерпретации текста в фигурных скобках. Процесс перевода опишем следующей декларацией:

```
(String) Expression
" [0-9]+ "
    { #1 } dif { '0' }
'x'
    { 'x' } dif { '1' }
'(' Expression ')'
    { '(' & %1 & ')' } dif { '(' & dif { %1 } & ')' }
'sin' '(' Expression ')'
    { 'sin' '(' & %1 & ')' } dif { 'cos' '(' & %1 & ')' * '(' & dif { %1 } & ')' }
'cos' '(' Expression ')'
    { 'cos' '(' & %1 & ')' } dif { '-sin' '(' & %1 & ')' * '(' & dif { %1 } & ')' }
'-' Expression
    { '-' & %1 } dif { '-' & dif { %1 } }
Expression '*' Expression
    { %1 & '*' & %2 } dif { '(' & %1 & '*' & dif { %2 } & '+' & dif { %1 } & '*' & %2 & ')' }
Expression "+" | "-" Expression
    { %1 & #1 & %2 } dif { '(' & dif { %1 } & #1 & dif { %2 } & ')' },
```

где  $\&$  — операция конкатенации строк (определена при описании String, в примере не показано),  $\#n$  — зарезервированный символ синтаксического анализатора, возвращающий строку, сопоставленную терминальному шаблону предложения с номером  $n$ , а  $\%m$  — символ, возвращающий строку, соответствующую сущности, обозначенной понятием с номером  $m$ .

Текст ситуационной части  $\text{'dif}\{\sin(5*\cos(x))-x*x\}$ , включающий в себя выражение, которое необходимо продифференцировать, будет переведен и представлен в виде следующего результата

$$\text{'(cos(5*cos(x))*(5*-sin(x)*(1)+0*cos(x)*(1))-(x*1+1*x)'}.$$

В случае необходимости, можно определить именованный императив  $\text{equ}$ , который выполнит очевидные тождественные преобразования выражений, получаемых после дифференцирования, например, путем задания ситуации  $\text{'equ}\{\text{dif}\{\sin(5*\cos(x))-x*x\}\}$ .♦

Реализация перевода атрибутными грамматиками сопряжена со значительными трудностями описания контекстно-зависимых условий или смысла конструкций языков программирования. Эти трудности связаны как с самим формализмом, так и с некоторыми технологическими проблемами [23]. К трудностям первого рода можно отнести неупорядоченность процесса вычисления выходных атрибутов при жесткой упорядоченности синтаксического анализа входного текста. Это несоответствие требует искусственных приемов для их сочетания. Технологические трудности определяются низкой эффективностью трансляторов, сгенерированных с помощью атрибутных систем, что связано с большим расходом памяти и наличием искусственных приемов итерационного вычисления атрибутов.

**Пример 11.** Рассмотрим пример реализации в контекстной технологии трансляции текстового представления числа в формате с фиксированной запятой в его двоичный эквивалент. Семантика такой задачи традиционно описывается атрибутной грамматикой.

Для выражения числа с фиксированной запятой Fixed введем два дополнительных понятия: Integer (целая часть) и Fraction (дробная часть). Понятию Integer припишем атрибут  $\text{Int}$ , равный значению целой части числа, а понятию Fraction — атрибут  $\text{Frac}$ , равный дробной части. Атрибуты сопоставим объектам-результатам предложений, выражающим соответствующие понятия:

```
(Arithmetic) Integer
"[0-9]" { #1 }
"[0-9]" Integer { %1 * 10 + #1 }
(Arithmetic) Fraction
"[0-9]" { #1 }
Fraction "[0-9]" { #1 + %1 / 10 }
(Integer) Fixed
Integer '.' Fraction { %1 + %2 / 10 },
```



где использовано не определенное в примере понятие Arithmetic. При декларации Arithmetic, очевидно, должны быть определены арифметические операции, которые использованы в императивах.

По терминологии атрибутивных грамматик атрибут Int является синтетическим, в то время как атрибут Frac — наследуемым. В отличие от атрибутивных грамматик, не имеющих средств задания порядка вычисления атрибутов, последовательность вычисления объектов в примере определена выразительными средствами самой модели. Для правильного вычисления синтезируемого атрибута понятие Integer определено как подлежащее грамматическому разбору слева направо. В свою очередь, для вычисления наследуемого атрибута понятие Fracition подвергается разбору справа налево. ♦

Заметим, что в понятийной модели с каждым понятием изначально ассоциируется атрибут, соответствующий обозначаемому объекту-результату. При необходимости, эти объекты можно описать как составные, т. е. содержащие требуемое число атрибутов.

### ЗАКЛЮЧЕНИЕ

Трудности, связанные с созданием и сопровождением современных информационных систем, в значительной мере объясняются семантическим разрывом, возникающим между содержательными представлениями о предметной области и решаемыми задачами и средствами языков программирования, служащими для решения этих задач. Для сокращения семантического разрыва предлагается применение контекстной технологии программирования, основанной на понятийном анализе предметной области и контекстной интерпретации лексем.

В контекстной технологии понятийный анализ используется для выявления понятийной структуры предметной области, отражаемой в понятиях создаваемого специализированного языка. Контекстная интерпретация, в свою очередь, позволяет естественным образом реализовать контекстные условия для специализированного языка и улучшить его выразительные возможности.

При использовании специализированного языка, создаваемого по контекстной технологии, ожидается получение более простых и надежных программ. Для оценки реальных улучшений и определения предпочтительных областей применения целесообразны экспериментальные исследования.

В заключение укажем на некоторые публикации, где приведены не вошедшие в настоящую статью описания особенностей контекстной технологии программирования. Так, в докладе [24] описан вычислительный механизм контекстной технологии и рассмотрен пример более сложной предметной области. В статье [25] содержится описание принципов контекстного программирования с их привязкой к вычислительной модели. Работа [26] посвящена вопросам создания, хранения и использования знаний, представленных в виде откомпилированных понятийных моделей, а также содержит описание адресных структур вычислительного механизма контекстной технологии.

### ЛИТЕРАТУРА

1. Марка Д. А., Мак-Гоуэн К. Методология структурного анализа и проектирования. — М.: Мета Технология, 1993.
2. Черемных С. В., Семенов И. О., Ручкин В. С. Структурный анализ систем: IDEF-технологии. — М.: Финансы и статистика, 2001.
3. Калайян А. Н., Калянов Г. Н. Структурные модели бизнеса: DFD-технологии. — М.: Финансы и статистика, 2003.
4. Конноли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. Теория и практика. — М.: Вильямс, 2003.
5. Meyer V. Object Technology: The Conceptual Perspective // Computer. — 1996. — N 1. — P. 86—88.
6. Буч Г. Объектно-ориентированное проектирование с примерами применения. — М.: Конкорд, 1992.
7. Рамбо Дж., Буч Г., Якобсон А. UML. Специальный справочник. — СПб.: Питер, 2002.
8. Александреску А. Современное проектирование на C++: Обобщенное программирование и прикладные шаблоны проектирования. — М.: Вильямс, 2004.
9. Выхованец В. С. Контекстная технология программирования // Труды IV Междунар. науч.-техн. конф. по телекоммуникациям (Телеком-99). — Одесса, 1999. — С. 116—119.
10. Хантер Р. Основные концепции компиляторов. — М.: Вильямс, 2002.
11. Горский Д. П. Вопросы абстракции и образования понятий. — М.: Изд-во АН СССР, 1961.
12. Рейгурд-Смит В. Теория формальных языков. Вводный курс. — М.: Радио и связь, 1988.
13. XML Schema Part 2: Datatypes. W3C Recommendation (<http://www.w3.org>).
14. Выхованец В. С. Язык контекстного программирования // Тез. докл. 8-й междунар. конференции “Проблемы управления безопасностью сложных систем”. — М., 2000. — Т. 2. — С. 89—91.
15. Макетирование, проектирование и реализация диалоговых информационных систем / Под ред. Е. И. Ломако. — М.: Финансы и статистика, 1993.
16. Гаскаров Д. В. Интеллектуальные информационные системы. — М.: Высш. шк., 2003.
17. Братчиков И. Л. Синтаксис языков программирования. — М.: Наука, 1975и.
18. Першиков В. И., Савинков В. М. Толковый словарь по информатике. — М.: Финансы и статистика, 1991.
19. Кнут Д. Семантика контекстно-свободных языков // Семантика языков программирования. — М.; 1980.
20. Шагурин И. И., Бердышев Е. М. Процессоры семейства Intel P6: Архитектура, программирование, интерфейсы: Спр. М.: Горячая линия — Телеком, 2000.
21. Пратт Т., Зелковиц М. Языки программирования: разработка и реализация. — СПб.: Питер, 2002.
22. Ахо А., Ульман Д. Теория синтаксического анализа, перевода и компиляции. — М.: Мир, 1978.
23. Бездушный А. Н., Лютый В. Г., Серебряков В. А. Разработка компиляторов в системе СУПЕР. — М.: ВЦ АН СССР, 1991.
24. Иосенкин В. Я., Выхованец В. С. Контекстная модель технологического процесса предприятия // Труды II междунар. конф. “Идентификация систем и задачи управления” SICPRO’03. — М., 2003. — С. 859—871.ц
25. Иосенкин В. Я. Контекстно-ориентированное программирование // Искусственный интеллект. — 2004. — № 3. — С. 667—677.
26. Выхованец В. С., Иосенкин В. Я. Компиляция знаний, представленных на языке ESSE // Тез. докл. 2-й Междунар. конф. по проблемам управления. — М., 2003. — Т. 2. — С. 165.

☎ (095) 787-88-44

E-mail: [vyk@ipu.ru](mailto:vyk@ipu.ru)



# ЯЗЫК ПАРСЕК: ПРОГРАММИРОВАНИЕ ГЛОБАЛЬНО РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ В МОДЕЛИ ИСЧИСЛЕНИЯ ДРЕВОВИДНЫХ СТРУКТУР

Ю. С. Затуливетер, А. В. Топорищев

*Институт проблем управления им. В.А. Трапезникова, г. Москва*

Для языка программирования ПАРСЕК, построенного в модели исчисления древовидных структур, разработаны функции управления глобально распределенными вычислениями. Рассмотрены особенности программирования в математически однородном поле компьютерной информации.

## ВВЕДЕНИЕ

Современные подходы к организации обработки и хранения распределенных данных в ресурсах глобальной Сети представлены grid-технологиями [1, 2], которые сформировались и апробировались в ходе решения сложных задач обработки больших объемов экспериментальных данных, получаемых в рамках крупномасштабных научных проектов ядерной физики, биологии и др. За неполное десятилетие испытаний из нескольких независимых подходов и систем наибольшее распространение получила инструментальная система Globus Toolkit (версии GT1, GT2) [3].

Актуальность проблематики распределенных вычислений в ресурсах Сети выходит далеко за рамки крупномасштабных научных проектов. В связи с особым значением задач консолидации в ресурсах Сети корпоративных информационных пространств grid-технологии переходят в фазу коммерциализации. В 2003 г. вышла третья версия Globus Toolkit (GT3), которая разрабатывалась в интересах бизнес-приложений при участии ведущих компьютерных компаний [3]. Перспективы быстрого развития grid-технологий связываются с растущей актуальностью проблем функциональной интеграции Сети для разнообразных задач в крупных корпорациях и ведомствах ([http://www.osp.ru/nets/2003/15/000\\_5.htm](http://www.osp.ru/nets/2003/15/000_5.htm)).

Между бизнес-приложениями и научными задачами, требующими распределенной обработки/хранения данных, имеется существенное отличие. Научные проекты — в большей своей части целевые и разовые. Средства распределенной обработки/хранения данных стационарно структурируются под ту или иную фундаментальную проблему и после завершения экспериментов утрачивают актуальность. В коммерческих применениях приложения существенно более динамичны и по структурам, и по требованиям реагирования и адаптации в реальном времени в ответ на быстрые изменения внешней среды.

Особое значение для бизнес-приложений имеют простота, доступность средств программирования с глобальной интеграцией распределенных данных, программ и систем не только со сложными, но и существенно нестационарными — быстро меняющимися — структурами.

“Появление масштабных инфраструктурных проектов обусловило переход к новой фазе развития программного обеспечения grid. Сыгравший свою роль Globus Toolkit перестает быть единственным доступным средством создания grid... Новое поколение программного обеспечения grid развивается в форме платформ... В связи с появлением нескольких альтернативных платформ на передний план выходит задача обеспечения interoperability grid разных виртуальных организаций, использующих различные исполнительные среды. Уже имеющийся опыт... показывает, что этого недостаточно: требуются определение базового состава сервисов и *унификация их семантики*” [3].

Многоплатформенность коммерческого grid-инструментария обостряет и без того крайне болезненную проблему неоднородности глобального компьютерного информационного пространства. Разрозненность форм представления и способов обработки компьютерной информации обуславливает один из фундаментальных факторов [4] торможения интеграционных процессов в глобальной компьютерной среде. Причины его появления уходят корнями в неадекватность классической компьютерной аксиоматики в модели фон Неймана [5] требованиям глобальной интеграции данных, программ, процессов и систем [4, 6–8].

Изначальная несовместимость различных платформ grid-технологий показывает, что общий подход, который бы исчерпывающим образом ответил на острые вопросы функциональной интеграции глобального информационного пространства в интересах сверхдинамично развивающейся социальной среды, в том числе





бизнес-среды, все еще не сложился. Компьютерная индустрия продолжает опираться на классические постулаты универсально программируемых вычислений, изначально предложенных для автономных компьютеров [5]. По этой причине ей до сих пор не удается выявить и устранить фундаментальные препятствия на пути к широкодоступным средствам программирования и интеграции глобально распределенных вычислений [6]. В эпоху глобальной Сети лидеры индустрии программирования под новыми флагами корпоративной консолидации глобального информационного пространства продолжают по старинке осуществлять политику продвижения своих платформ, каждая из которых заведомо не пригодна к полномасштабной функциональной интеграции компьютерной среды, поскольку в своих логических основаниях опирается на постулаты автономного компьютера.

При этом все еще отсутствует единое стратегическое понимание главной общей задачи grid-технологий. Весьма показательно мнение руководителя группы баз данных CERN Джейми Шиерса — одного из самых опытных специалистов в сфере применения grid к сложным задачам. На вопрос “Что, по Вашему мнению, может стать “убойным” приложением для появления действительно глобальной, всепроникающей grid-среды?” (<http://www.osp.ru/os/2004/12/034.htm>) он ответил: “Я много думаю над этим. Крайне сложно предсказать заранее. Возможно, это приложение, которое сейчас повсеместно используется, но мы просто не видим его потенциала. Но это должно быть нечто, что действительно изменит нашу жизнь, как Internet, который мы используем всегда и везде. *Это не может быть задачей из какой-либо частной области, это должно быть приложение, которое реально повлияет на стиль нашей повседневной работы.* Честно говоря, пока я такого приложения не вижу. Видимо, нам еще предстоит его найти”. Курсивом мы выделили ключевую, как нам представляется, мысль, указывающую правильное направление поиска такой задачи.

В работах [7, 8] сделана попытка дать максимально общее определение задачи, которая составит единое представление о целях и особенностях глобально распределенной обработки данных. Это задача кибернетизации в математически однородном поле компьютерной информации социосистемы в целом. Все решаемые с помощью глобальной Сети задачи (экономика, госуправление, здравоохранение, природопользование, наука и другие) так или иначе сводятся к процессам компьютерной кибернетизации социальной среды через все ее институты самообеспечения, саморегулирования и самоорганизации. В общем случае это непрерывный спектр компьютерных задач установления системных балансов, направленных на обеспечение управляемого развития (“sustainable development”) человечества<sup>1</sup>.

Показано [7, 8], что для решения общей задачи компьютерной кибернетизации социосистемы требуется наличие адекватного компьютерного инструментария для программирования глобально распределенных вычислений и процессов управления. К ключевым требо-

ваниям следует отнести способность к глобальной интеграции распределенных данных, программ, процессов и систем, а также простоту и широкую доступность средств программирования распределенных задач высокой структурной сложности.

Особо подчеркнем, что для создания инструментария полномасштабного решения и интеграции глобально распределенных задач в ресурсах Сети требуется переход от классических постулатов универсальных вычислений в автономных компьютерах к новой аксиоматике математически однородного поля компьютерной информации [4, 6–8].

В качестве прототипа, отвечающего этим требованиям, мы рассматриваем язык и систему программирования ПАРСЕК (ParSeq) [9], в которых воплощена аксиоматика математически однородного поля компьютерной информации, семантика которого определяется математически замкнутой моделью исчисления древовидных структур.

Цель построения и развития языка и системы программирования ПАРСЕК — формирование подхода к программированию в аксиоматике математически однородного поля компьютерной информации (программ и данных), которая открывает практически неограниченные возможности глобальной интеграции данных, программ, вычислительных процессов и систем в рамках математически замкнутых формальных моделей.

## 1. КРАТКОЕ ОПИСАНИЕ ЯЗЫКА ПАРСЕК

Язык ПАРСЕК [9] является процедурной реализацией исчисления древовидных структур — математически замкнутой модели вычислений на множестве вполне структурированной информации. Области применения языка и системы программирования ПАРСЕК — разнообразные задачи высокой структурной и динамической сложности, предназначенные для компьютерной реализации. Вычислительная среда — ресурсы отдельного компьютера (система реализована на платформе win32), ресурсы локальных сетей и глобальная компьютерная среда (система в стадии разработки).

### 1.1. Особенности языка

Отметим некоторые важные особенности модели исчисления древовидных структур и языка ПАРСЕК. Отличительное свойство языка ПАРСЕК — сочетание математической замкнутости с процедурностью. Программирование осуществляется посредством замкнутого и полного набора операций с *древовидными и только древовидными структурами*, составляющего суть исчисления древовидных структур. При этом обеспечены следующие ключевые для широкого использования качества:

- система остается универсальной;
- математическая замкнутость исчисления древовидных структур позволила в процедурном исполнении построить автоматическое управление внутренними вычислительными ресурсами компьютера;
- система изолирует программиста от сложностей управления вычислительной средой и позволяет ему решать задачи в формальных математических конструкциях, не связанных с особенностями машинной среды.

Важно отметить, что исчисление древовидных структур вполне допускает расширение автоматического уп-

<sup>1</sup> Концепция “sustainable development” была сформулирована как стратегическая линия развития человечества на Конференции ООН по окружающей среде и развитию (ЮНСЕД) 1992 г., Рио-де-Жанейро.

правления вычислительными ресурсами с внутренних ресурсов одного компьютера до ресурсов внешних компьютеров, связанных между собой сетями. Такое свойство дает формальную основу для полнофункциональной интеграции вычислительных ресурсов локальных сетей и глобальной компьютерной среды в целом с формированием в ней математически однородного поля компьютерной информации.

Первая версия языка и системы программирования ПАРСЕК была построена и испытана на задачах в 1993–1994 гг. С тех пор система ПАРСЕК развивается в трех своих компонентах: системное окружение, библиотеки встроенных функций и классы решаемых задач.

В работе представлены результаты текущего этапа развития системы ПАРСЕК во всех трех компонентах. Оно направлено на расширение сфер применимости системы ПАРСЕК с вычислительных ресурсов отдельного компьютера на распределенные ресурсы сетей.

Описание языка ПАРСЕК приводится по тексту [9].

### 1.2. Первичные понятия

При описании информационных объектов и систем используются формулировки типа: *A* состоит из *a*, *b* и *c*. В теории множеств этой формулировке соответствует дефиниция  $A = \{a, b, c\}$ . В языке ПАРСЕК для “раскрытия” объекта используется элементарное дерево в геометрической форме:

```

┌A
├a
├b
└c
    
```

Дерево состоит из вершин. Вершина характеризуется содержимым вершины и связями с другими вершинами. Имеются три вида связей для каждой вершины:

- *deep* — связь с подчиненной по уровню вершиной;
- *next* — связь с нижеследующей вершиной одного уровня;
- *prev* — связь с предыдущей вершиной.

Пример дерева и анализ его структуры приведен в табл. 1.

Двоичный набор из признаков наличия или отсутствия каждого из трех видов связей (*prev*, *deep*, *next*) образует статус вершины. Статусы вершин (связи дерева) отображаются с помощью псевдографических символов T, t, L, а также вспомогательного символа |, который дополняется в левой части строк, несущих изображение вершин старшего ранга.

Древовидная структура представляет те или иные вполне структурированные информационные объекты. Обработка объекта ведется последовательно по компо-

нентам. Для выделения компонентов, которыми являются вершины, вводится понятие *курсор*.

*Курсор* — это переменная, которая принимает значение адреса (ссылки, указателя) некоторой вершины. На дереве можно определить один или более одного курсора. Процесс обработки начинается с открытия курсоров и перемещения их по дереву.

С курсором связан базисный набор функций для работы с деревьями. Они разделяются на следующие группы.

#### Одношаговое перемещение курсора

*deep()* — переход на подчиненную вершину,

форма вызова:  $cur2 = deep(cur1)$ , где *cur1* — адрес текущей вершины, *cur2* — адрес вершины, подчиненной текущей;

*next()* — переход на нижнюю вершину того же ранга,

форма вызова:  $cur2 = next(cur1)$ ;

*prev()* — переход на предыдущую вершину,

форма вызова:  $cur2 = prev(cur1)$ .

#### Функции анализа и замены содержимого

*cont()* — прочитать содержимое вершины,

форма вызова:  $c = cont(cur)$  или  $c = \{cur\}$ , где *c* — содержимое вершины, *cur* — адрес вершины (курсор);

*re\_cont()* — занесение нового содержимого вершины,

форма вызова:  $re\_cont(cur, NC)$  или  $\{cur\} = NC$ ,

где *cur* — адрес вершины, *NC* — новое содержимое.

#### Создание и уничтожение вершин

*node\_create()* — создать вершину,

форма вызова:  $cur = node\_create(NC)$ , где *cur* — переменная (с произвольным именем), которая является вновь создаваемым курсором, указывающим на созданную вершину, *NC* — новое содержимое, каждая создаваемая вершина является изолированной;

*node\_del()* — уничтожить вершину,

форма вызова:  $node\_del(cur)$ , где *cur* — текущий курсор.

#### Редактирование связей

*link\_next( , )* — связать две вершины на одном уровне;

*link\_deep( , )* — связать две вершины так, чтобы вторая стала подчиненной первой;

*link\_del( , )* — разорвать связь между вершинами,

формы вызова:  $link\_next(cur1, cur2)$ ,  $link\_deep(cur1, cur2)$ ,  $link\_del(cur1, cur2)$ , где *cur1* — адрес первой вершины, *cur2* — адрес второй вершины.

### 1.3. Структура программы

Структура ПАРСЕК-программ вполне классическая и типична для процедурного программирования.

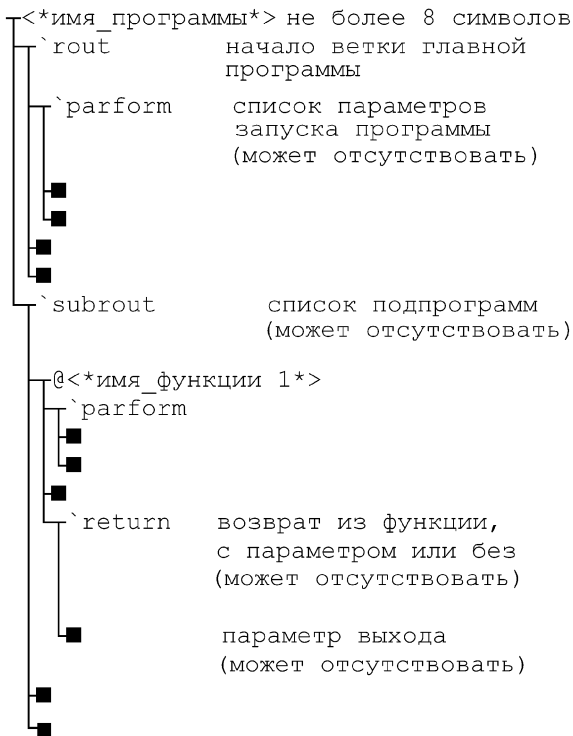
Таблица 1

Определение структуры двоичного дерева

Дерево	Номер вершины	Содержимое вершины	Связь prev	Связь deep	Связь next
┌OBJ	1	OBJ	—	С вершиной 2	—
├A	2	A	С вершиной 1	С вершиной 3	С вершиной 5
├-a1	3	a1	С вершиной 2	—	С вершиной 4
├-a2	4	a2	С вершиной 3	—	—
└B	5	B	С вершиной 2	—	—



Особенность лишь в том, что тексты программ представляются в виде двоичного дерева с ключевыми словами:



Символьные строки в правой части вершины дерева после первого пробела образуют комментарий (например: строка “начало ветки главной программы”).

Условные обозначения:

- ■ — поддерево или лист;
- ключевые слова в языке ПАРСЕК начинаются с символа “`” (обратный апостроф), т. е. <\*ключевое\_слово\*> = `<\*идентификатор\*>, с помощью ключевых слов строятся все конструкции языка;
- идентификатор встроенных функций начинается с символа “\_”, пользовательских — с символа “@”, т. е. <\*встроенная\_функция\*> = \_<\*идентификатор\*>, <\*пользовательская\_функция\*> = @<\*идентификатор\*>.

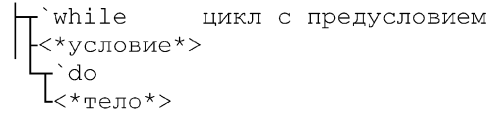
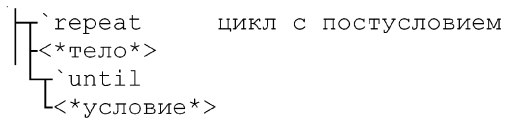
Таблица 2

**Представление операций и выражений**

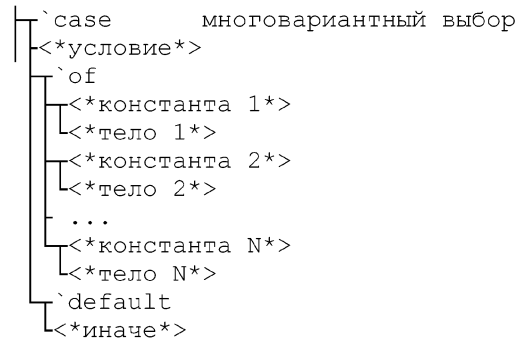
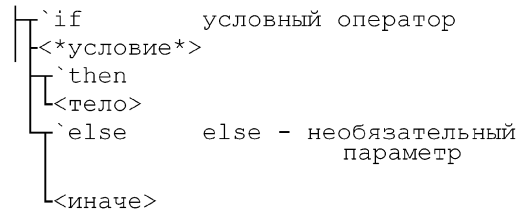
Конструкция	Формальный вид	Древовидная запись
Присваивание	$a = b$	$\begin{matrix} \top \\ \text{a} \\ \text{b} \end{matrix}$
Операция	$1 + 3$	$\begin{matrix} \top \\ \text{+} \\ \text{1} \\ \text{3} \end{matrix}$
Функция	$f_1(x, y) \quad @f_2(x, y)$	$\begin{matrix} \top \\ \text{-f1} \\ \text{x} \\ \text{y} \end{matrix} \quad \begin{matrix} \top \\ \text{-@f2} \\ \text{x} \\ \text{y} \end{matrix}$
Выражение	$a = 1 + 3$	$\begin{matrix} \top \\ \text{a} \\ \text{=} \\ \text{1} \\ \text{+} \\ \text{3} \end{matrix}$

Базовые конструкции языка ПАРСЕК вполне традиционны для процедурного программирования:

**Циклы**



**Условия**



Операция присваивания, выражения и функции в древовидной записи представлены в табл. 2.

**2. АРХИТЕКТУРА РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ**

Информационное пространство WWW построено по гипертекстовой модели в сетевой архитектуре “клиент—сервер”. Модель гипертекста не является моделью универсальных вычислений, поэтому изначально не поддерживает распределенные вычисления в ресурсах глобальной компьютерной среды.

Проблемы организации универсально программируемых распределенных вычислений в сетевой архитектуре P2P (Peer-to-Peer), в которых вычислительные и управляющие задачи могли бы реализовываться в виде распределенных параллельных процессов в ресурсах глобальной компьютерной среды, оказываются принципиально более сложными, чем в локальных сетях. Инструментарий параллельного программирования распределенных вычислений в локальных сетях и кластерах (PVM, MPI и др.) не обладает свойством произвольной масштабируемости в глобально распределенных вычислительных ресурсах. В настоящее время индустриальное программирование не имеет регулярных, легко масштабируемых средств программирования глобально распределенных вычислений и процессов, способных обеспечить массовые потребности в решении глобальных задач [6—8]. Глобальные задачи — это новый класс задач, решение которых предполагает согласованное использова-

ние (в автоматических режимах) вычислительных ресурсов компьютеров (в пределе — всех) глобальной Сети.

Стратегическая задача развития глобальной компьютерной среды состоит в структурной и функциональной интеграции ее вычислительных и информационных ресурсов, которая позволит всем компьютерам согласованно работать над одной или многими задачами в составе единого универсально программируемого глобального компьютера [6—8, 10, 11].

В связи с нарастающей глобализацией социальных процессов проблемы универсально программируемой распределенной обработки в глобально распределенных вычислительных ресурсах выходят на высшие приоритеты компьютерной индустрии [7, 8]. Их актуальность увеличивается по двум взаимосвязанным причинам.

Во-первых, из-за глобализации мировой экономики растет спрос на решение глобальных задач обработки и хранения данных в ресурсах Сети, которые отличаются высокой структурной и динамической сложностью, требуют переработки в реальном времени сверхбольших потоков разнообразной информации, обладающий свойствами глобальной связности.

Во-вторых, современные средства индустриального программирования, ориентированные на решение локализуемых задач, оказались не готовыми к требованиям глобальных задач, которые играют все более важную роль в глобальном информационном пространстве.

В данной работе развивается подход [4, 6—12] к реализации глобально распределенных вычислений в сетевой архитектуре P2P в вычислительной модели исчисления древовидных структур, реализованного в системе программирования ПАРСЕК [9]. Для исследования возможностей программирования в ресурсах глобальной компьютерной среды в системе ПАРСЕК строятся функции и среда программной эмуляции распределенных вычислений в ресурсах глобальной компьютерной среды. Архитектура среды такой эмуляции представлена далее.

### 2.1. Функционально полный набор действий по управлению распределенными ресурсами

В сетевой архитектуре P2P многие пары компьютеров связаны непосредственно между собой так, что в целом образуют *связную сеть*.

На рис. 1 показана отдельная пара связанных между собой компьютеров. Каждый из этих компьютеров может быть связан и с другими компьютерами. Конкретная топология таких сетей для дальнейшего рассмотрения несущественна.

Каждый компьютер сети имеет свой адрес, состоящий из двух компонентов: IP-адрес и IP-порт (в вырож-

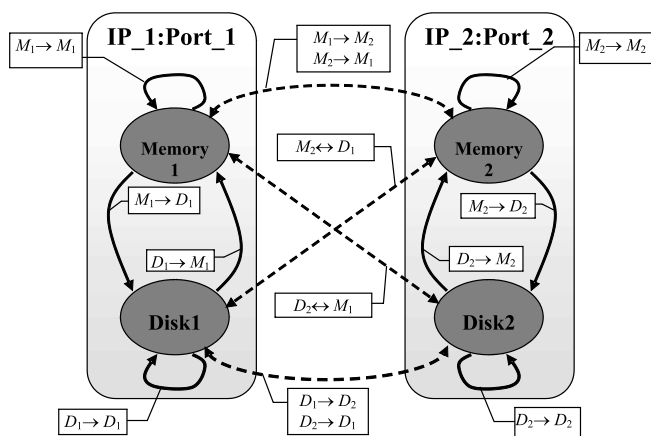


Рис. 1. Произвольная пара взаимосвязанных компьютеров с указанием направления действий (из  $X \rightarrow Y$ ) между разными видами памяти

денном случае сеть может быть “организована” на одном компьютере с адресом IP, но с разными портами). В каждом компьютере имеется два вида ресурсов памяти: оперативная память и дисковая память (на рисунке **Memory1**, **Disk1** и **Memory2**, **Disk2**, соответственно).

Стрелки с надписями показывают направления действий из одной памяти в другую. Оперативная память разных компьютеров обозначена буквами  $M_1$  и  $M_2$ , дисковая — буквами  $D_1$  и  $D_2$  (первого и второго компьютеров, соответственно). В табл. 3 перечислены все направления и типы действий, показанные на рис. 1 стрелками.

В дисковой памяти, находящейся под управлением файловой системы, деревья хранятся в виде символьных файлов.

В оперативной памяти деревья представлены в виде внутренних структур. В языке ПАРСЕК функции ввода/вывода (`_read_tree/ write_tree`) обеспечивают преобразование деревьев, хранящихся в файлах, во внутренние структуры и наоборот.

В табл. 4 приведены все варианты адресации к внутри- и межкомпьютерным ресурсам памяти.

Одна из главных целей расширения системы ПАРСЕК на распределенные вычислительные ресурсы сетей — это достижение независимости программ от расположения древовидных структур в ресурсах всех компьютеров, связанных сетями и являющихся носителями математически однородного поля компьютерной информации.

Таблица 3

Возможные типы действий с использованием всех видов памяти

Внутрикомпьютерные из $X_i \rightarrow Y_i, i = 1, 2$			Межкомпьютерные (через сеть) из $X_i \rightarrow Y_j, i \neq j, i = 1, 2$		
$M_i \rightarrow M_i$	$D_i \rightarrow M_i,$ $M_i \rightarrow D_i$	$D_i \rightarrow D_i$	$M_i \rightarrow M_j$	$M_i \rightarrow D_j,$ $D_i \rightarrow M_j$	$D_i \rightarrow D_j$
Чтение/запись элементов дерева по адресу, обработка, присваивание	Ввод (read), вывод (write)	Управление файлами (create, del, copy, move, rename,...)	Чтение/запись элементов дерева по адресу, обработка, присваивание	Ввод (read), вывод (write)	Управление файлами (create, del, copy, move, rename,...)





Первый шаг в этом направлении — расширение областей применимости операций, функций и выражений языка ПАРСЕК с внутрикомпьютерных ресурсов оперативной и дисковой памяти на межкомпьютерные. При этом в ходе исполнения перечисленных действий обработка элементов деревьев должна осуществляться с автоматической локализацией мест расположения элементов деревьев, к которым они применяются.

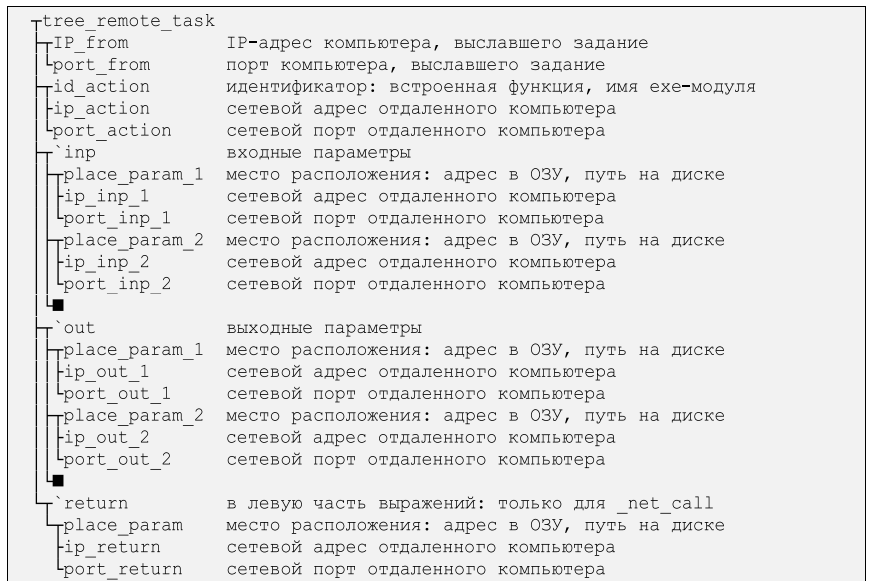
В табл. 5 приведены инвариантные (относительно размещения элементов деревьев в ресурсах разных компьютеров) операторы всех трех видов действий.

Таким образом, рис. 1 и табл. 3—5 дают представление о функционально полном наборе функций управления распределенными ресурсами и принципах расширения действий языка ПАРСЕК с деревьев в ресурсах отдельного компьютера на ресурсы всех связанных компьютеров, являющихся носителями математически однородного поля компьютерной информации.

**2.2. Структура распределенной обработки**

Распределенная обработка осуществляется двумя способами управления обработкой деревьев, компоненты которых распределены по разным компьютерам. Первый — последовательная обработка компонентов деревьев, распределенных по разным компьютерам. Второй — параллельная обработка компонентов деревьев, распределенных по разным компьютерам посредством одновременного исполнения на разных компьютерах многих асинхронно взаимодействующих процессов.

Первый способ реализуется последовательным вызовом функции `_net_call(tree_remove_task)`. Дерево `tree_remove_task` — параметр, представляющий задание для исполнения на отдаленном компьютере (рис. 2).



**Рис. 2. Структура дерева заданий для исполнения на отдаленных компьютерах**

Знак ■ на рис. 2 показывает, что число входных и выходных параметров не фиксируется (от нуля и более). Ветка `return` может использоваться для функции `_net_call()`.

Второй способ реализуется с помощью основной пары функций `net_pid = net_start(tree_remote_task)` и `_net_wait(net_pid)`, аналогами которых являются хорошо известные функции запуска процессов (`pid = start_process()`) с формированием `pid`-идентификаторов открытых процессов и ожидания их завершения (`wait(pid)`). С их помощью в существующих операционных системах осуществляется асинхронное управление многопроцессной обработкой.

В реализации многопроцессной обработки в системе ПАРСЕК добавлены также сопутствующие функции уп-

Таблица 4

**Варианты адресации к ресурсам памяти**

Область действия адресации	Виды памяти	Адресация к разным видам памяти	Форматы адресации	Примеры предоставления адресации деревом
Внутрикомпьютерная, $i = 1, 2$	$M_i$	adr_memory	Слово 32—64 (или более) двоичных разрядов	<code>'adr</code> <code>\$123456789ABCDEF</code>
	$D_i$	path_disk	Символьная строка-путь	<code>'path</code> <code>c:/dir_1/file</code>
Межкомпьютерная (через сеть), $j = 1, 2, j \neq i$	$M_i$	adr_memory_net	ip:port#adr_memory	<code>'net</code> <code>ip</code> <code>port</code> <code>adr</code> <code>\$123456789ABCDE</code>
	$D_j$	path_disk_net	ip:port/path_disk	<code>'net</code> <code>ip</code> <code>port</code> <code>path</code> <code>c:/dir_1/file</code>

**Инвариантные операторы трех видов действий**

Виды действий	Инвариантные относительно размещения элементов деревьев операторы (номера компьютеров $i, j = 1, 2$ )	$Y_j$ (ресурс для результирующих операндов)	$X_i$ (ресурс для исходных операндов)	Область действия: $i = j$ – внутри компьютера; $i \neq j$ – между компьютерами
Присваивание, обработка	$\sim\text{Sign} (Y_j, X_i), \sim\text{Exec} (Y_j, X_i)$	adr_memory	adr_memory	$i = j$
			adr_memory_net	$i \neq j$
		adr_memory_net	adr_memory	$i \neq j$
			adr_memory_net	$i \neq j$
Ввод-вывод деревьев	$Y_j = \_Read\_tree (X_i)$	adr_memory	path_disk	$i = j$
			path_disk_net	$i \neq j$
		adr_memory_net	path_disk	$i \neq j$
			path_disk_net	$i \neq j$
	$\_Write\_tree (X_i, Y_j)$	path_disk	path_disk	$i = j$
			path_disk_net	$i \neq j$
		path_disk_net	adr_memory	$i \neq j$
			adr_memory_net	$i \neq j$
Управление файлами	$\_Copy (X_i, Y_j)$	path_disk	path_disk	$i = j$
			path_disk_net	$i \neq j$
		path_disk_net	path_disk	$i \neq j$
			path_disk_net	$i \neq j$
	$\_Move (X_i, Y_j)$	path_disk	path_disk	$i = j$
			path_disk_net	$i \neq j$
		path_disk_net	path_disk	$i \neq j$
			path_disk_net	$i \neq j$
	$\_Rename (X_i, Y_j)$	path_disk	path_disk	$i = j$
			path_disk_net	$i \neq j$
		path_disk_net	path_disk	$i \neq j$
			path_disk_net	$i \neq j$
	$\_Del (X_i)$	нет	path_disk	$i = j$
			path_disk_net	$i \neq j$

Примечание. Операторы  $\sim\text{Sign} (,)$ ,  $\sim\text{Exec} (,)$  понимаются как обобщение всех действий языка ПАРСЕК, связанных с обработкой деревьев в оперативной памяти.

правления асинхронными процессами — уничтожение процесса, приостановка, диагностика текущих состояний, но в данной работе они не обсуждаются.

### 3. ПРИНЦИПЫ РЕАЛИЗАЦИИ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ

Распределенная обработка состоит из двух стадий: конфигурирования системы и исполнения заданий.

На стадии конфигурирования на всех компьютерах, связанных в сеть распределенной обработки, запускают-

ся ОЗУ-резидентные модули программы приемника данных. Они обеспечивают прием заданий через указанный в параметрах запуска сетевой порт, а также возврат результата.

Стадия исполнения каждого задания состоит из следующих этапов:

- подготовка (оформление) задания (в формате, представленном на рис. 2) и вызов либо функцией `_net_call`, либо `_net_start`;
- доставка задания на отдаленный компьютер;
- исполнение задания на отдаленном компьютере;



- доставка результатов к местам, указанным в задании на обработку.

Реализация стадии исполнения подробнее рассматривается ниже.

### 3.1. Резидентные процессы приема заданий

На каждой машине сети, вовлеченной в процесс распределенной обработки, резидентно работают, как минимум, два процесса: приемник заданий (PS\_Receiver) и интерпретатор команд языка ПАРСЕК (PSi\_net).

Приемники заданий обрабатывают вычислительные запросы. Они связаны с одним или несколькими системными серверами управления доступными вычислительными ресурсами, которые следят за наличием свободных компьютеров. На стадии конфигурирования распределенной системы приемник сразу после запуска соединяется с одним из системных серверов и передает ему доступную для распределенной системы обработки конфигурацию компьютера, на котором он запущен, открывая, тем самым, доступ к выделенным вычислительным ресурсам своего компьютера. Вместе с этим приемник инициализирует системные переменные и буферы ОЗУ, необходимые ему для приема и запуска заданий, а также запускает интерпретатор языка ПАРСЕК.

При получении заданий от удаленных компьютеров приемник определяет тип задания и передает его на обработку в интерпретатор. Компьютеру, инициировавшему выполнение задания, возвращается сообщение о его завершении и, если указано в задании, результат обработки.

Интерпретатор служит для обработки ПАРСЕК-команд, передаваемых ему приемником в задании. Задания передаются интерпретатору через буфер общей памяти с зарезервированным именем PS\_RCV\_SHARED\_NAME. Передача данных защищена мьютексом с зарезервированным именем PS\_RCV\_SHARED\_MUTEX и осуществляется по событию с зарезервированным именем PS\_RCV\_SHARED\_EVENT.

### 3.2. Функция передачи данных из пользовательских процессов

Передача данных или заданий удаленному компьютеру производится следующей функцией: `kod_z = net_send(ip, port, string, len_string)`, где `ip` — IP-адрес удаленного компьютера, в который передаются данные, `port` — IP-порт удаленного компьютера, `string` — указатель на передаваемые данные в виде двоичной строки, `len_string` — размер передаваемой строки данных, `kod_z` — код завершения работы функции (0 — во время передачи произошла ошибка, 1 — в буфере `string` находится результат обработки отправленных данных, 2 — завершение без возвращения результата).

В общем случае строка `string` произвольного размера. В текущей версии размер строки не превышает размера буфера “общей памяти” (128 К) с зарезервированным именем PS\_RCV\_SHARED\_NAME. Для отправки строки функцией передатчика буфер “общей памяти”, аналогичный тому, который открывается в приемнике, не требуется. Передающая программа не требует реализации в виде резидентного процесса (двух

процессов, между которыми нужен буфер обмена данными, просто нет).

С помощью передающей функции `_net_send()` реализуются функции `_net_call()` и `net_start()`.

Вызов функции последовательного запуска действий на удаленных компьютерах осуществляется в пользовательских программах и имеет следующий вид: `rez = net_call(tree_remote_task)`, где `rez` — результат: либо дальняя (сетевая) адресная ссылка, либо двоичная строка, либо дерево (курсор корневой вершины), `tree_remote_task` — дерево подготавливаемого задания (см. рис. 2).

Запуск и синхронизация (ожидание завершения) параллельных процессов на удаленных компьютерах осуществляется в пользовательских программах следующим образом.

Запуск:

```
net_pid_1 = net_start(tree_remote_task_1)
net_pid_2 = net_start(tree_remote_task_2)
...
```

Ожидание завершения:

```
_net_wait(net_pid_1)
_net_wait(net_pid_2)
...
```

Эти операторы реализуются средствами ПАРСЕК с помощью функции `_net_send(ip, port, string_task, len_string_task)`.

Здесь `net_pid_1`, `net_pid_2`, ... — уникальные идентификаторы процессов открываемых на удаленных компьютерах, `tree_remote_task_1`, `tree_remote_task_2`, ... — деревья с указанием обратных адресов, действий, операндов их сетевых адресов (см. рис. 2).

### 3.3. Прием-исполнение заданий

На передающей стороне в пользовательских программах осуществляется строго последовательное обращение к передающей программе из пользовательского процесса, что обеспечивается вызовом функции `_net_send()` и завершением ее работы сразу после завершения передачи. Резидентные процессы на каждом принимающем компьютере работают следующим образом.

Процесс приемника PS\_receiver создает (при запуске) или уничтожает (при закрытии) буфер “общей памяти” с зарезервированным именем PS\_RCV\_SHARED\_NAME, создает (при запуске) или уничтожает (при закрытии) событие с именем PS\_RCV\_SHARED\_EVENT, принимает из входного порта строку с данными или заданием, передает ее через буфер PS\_RCV\_SHARED\_NAME в процесс интерпретатора PSi\_net. После декодирования и исполнения задания интерпретатор возвращает сигнал завершения. Сигнал завершения может нести с собой результат обработки — строку с адресной ссылкой или с деревом результата, которые возвращаются из приемника передающей функции в компьютер-источник.

Процесс интерпретатора PSi\_net обрабатывает деревья, передаваемые ему в строке, принятой приемником, через буфер “общей памяти”. Связь приемника PS\_receiver с интерпретатором PSi\_net через буфер “общей памяти” с зарезервированным именем PS\_RCV\_SHARED\_NAME обеспечивает:

- передачу принимаемой строки через буфер в процесс PSi\_net:
  - целиком (для строк, размером не больших буфера);
  - последовательно по частям (для строк, размером больших буфера);
- возврат процессу — приемнику PS\_receiver строки результата:
  - целиком (для строк, размером не больших буфера);
  - последовательно по частям (для строк, размером больших буфера).

Синхронизация и корректность передачи данных через буфер обеспечивается событием с именем PS\_RCV\_SHARED\_EVENT.

### ЗАКЛЮЧЕНИЕ

Перспективные области применения предложенного подхода — решение задач глобально распределенной обработки информации, связанных с кибернетизацией социосистемы в целом [7, 8, 10, 11]. Имеются области, в которых создание высокоэффективных, безопасных и свободно масштабируемых глобально распределенных систем управления актуальны уже сейчас. Это — разнообразные сферы производства — потребления, а также госуправления (создание общегосударственных систем мониторинга и управления в сферах здравоохранения, социальной защиты, образования, науки и др. [13]). Полномасштабное решение многих глобально распределенных задач управления пока невозможно из-за отсутствия адекватного компьютерного инструментария.

Для глобализации функционального пространства компьютерной среды требуется единая модель универсально программируемых распределенных вычислений, сочетающая в себе следующие качества, распространяемые на глобальную компьютерную среду:

- структурная сложность форм компьютерного представления данных и программ должна быть минимальной;
- модель не должна допускать стихийного воспроизводства в компьютерной среде разнородных, плохо совместимых форм представления данных, программ, процессов и систем;
- модель должна обеспечить простоту программирования, целостность и свободную масштабируемость глобально распределенной обработки информации.

Из-за отсутствия модели, отвечающей перечисленным требованиям, компьютерная среда оказалась неготовой к общему решению проблем глобализации распределенных вычислений.

Исчисление древовидных структур, реализованное в системе программирования ПАРСЕК, может рассматриваться как основа для построения практически значимой реализации универсальной модели распределенной обработки глобально распределенных данных, которая открывает пути к практическому воплощению перечисленных свойств.

Данная работа является одним из первых практических шагов к созданию простого и широко доступного компьютерного инструментария для воплощения в компьютерной среде глобально распределенных процессов управления, которые обеспечат практическое

воплощение структурно сложных задач управления, решаемых с помощью математического аппарата, развивающего идеи графодинамики [14].

### ЛИТЕРАТУРА

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of High Performance Computing Applications. — 2001. — 15 (3) ([www.globus.org/research/papers/anatomy.pdf](http://www.globus.org/research/papers/anatomy.pdf)).
2. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration / I. Foster, C. Kesselman, J. Nick, S. Tuecke (<http://www.globus.org/research/papers/ogsa.pdf>).
3. Коваленко В., Корягин Д. Организация grid: есть ли альтернативы? // Открытые системы. — 2004. — № 12 (<http://www.osp.ru/os/2004/12/034.htm>).
4. Затуливетер Ю. С. Информация и эволюционное моделирование // Труды междунар. конф. “Идентификация систем и задачи управления” SICPRO’2000, Москва, 26—28 сентября 2000 г. Ин-т проблем управления РАН. — М., 2000. — С. 1529—1573 ([http://zvt.hotbox.ru/1529\\_.htm](http://zvt.hotbox.ru/1529_.htm)).
5. Беркс А., Голдстейн Г., Нейман Дж. Предварительное рассмотрение логической конструкции электронного вычислительного устройства // Кибернетический сборник. — 1964. — Вып. 9. — С. 7—67.
6. Затуливетер Ю. С. На пути к глобальному программированию // Открытые системы. — 2003. — № 3. — С. 46, 47 (<http://www.osp.ru/os/2003/03/046.htm>).
7. Затуливетер Ю. С. Проблемы глобализации парадигмы управления в математически однородном поле компьютерной информации. Ч. 1. Кибернетизация социосистемы // Проблемы управления. — 2005. — № 1. — С. 2—10 (<http://www.ipu.ru/period/pu/docs/zatul.pdf>).
8. Затуливетер Ю. С. Проблемы глобализации парадигмы управления в математически однородном поле компьютерной информации. Ч. 2. К единому функциональному пространству // Там же. — 2005. — № 2. С. 12—23.
9. Затуливетер Ю. С., Халатян Т. Г. ПАРСЕК — язык компьютерного исчисления древовидных структур с открытой интерпретацией. Стеновый вариант системы программирования. — М.: Ин-т проблем управления РАН, 1997. — 71 с.
10. Затуливетер Ю. С. Предпосылки появления глобального управляющего компьютера // Датчики и системы. — 2005. — № 4. С. 53—64.
11. Затуливетер Ю. С. Пути реализации глобального управляющего компьютера // Там же. — № 5. — С. 25—33.
12. Затуливетер Ю. С. Компьютерная информация в модели исчисления древовидных структур // Труды второй междунар. конф. “Идентификация систем и задачи управления” SICPRO’2003, Москва, 29—31 января 2003 г. / Ин-т проблем управления РАН. — М., 2003. — С. 790—858.
13. Концепция использования информационных технологий в деятельности федеральных органов государственной власти до 2010 года (<http://www.it-gov.ru/site.shtml?id=76>).
14. Динамический подход к анализу структур, описываемых графами (Основы графодинамики). I, II / М. А. Айзерман, Л. А. Гусев, С. В. Петров, И. М. Смирнова // Автоматика и телемеханика. — 1977. — № 7. — С. 135—151; № 9. — С. 123—136.

☎ (095) 334-92-09

E-mail: [zvt@ipu.rssi.ru](mailto:zvt@ipu.rssi.ru)

E-mail: [Artur.Toporischev@idm.ru](mailto:Artur.Toporischev@idm.ru)







# ОСОБЕННОСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СЛОЖНЫХ ИНТЕГРИРОВАННЫХ АСУТП

А. Г. Полетыкин

*Институт проблем управления им. В. А. Трапезникова, г. Москва*

Описаны основные задачи интегрирующего программного обеспечения (ПО), приведены типовые решения. Обсуждены основные трудности и проблемы, возникающие при интеграции разнотипного оборудования и ПО. Дан анализ опыта разработки ПО системы верхнего блочного уровня АСУТП для АЭС.

## ВВЕДЕНИЕ

Сложные интегрированные АСУТП все шире применяются в промышленности. Необходимость в них возникает, когда объект автоматизации содержит большое число датчиков и механизмов, а особенности технологического процесса не позволяют разбить его на набор слабосвязанных подсистем, которые могут контролироваться и управляться отдельными АСУТП. Такие интегрированные АСУТП применяются для управления ядерными объектами, в частности, АЭС.

В подавляющем большинстве существующих интегрированных АСУТП, включая АСУТП большинства российских АЭС, проблема интеграции средств контроля и органов управления решается путем их размещения на единых пультах управления. Этот способ обладает своими достоинствами и недостатками, обсуждение которых выходит за рамки настоящей статьи.

С наступлением эпохи массового применения микропроцессорной техники разработчики АСУТП стали решать проблему интеграции с помощью компьютеризированных рабочих мест. В качестве удачного примера такого решения можно назвать компьютеризированный блочный пульт управления промышленным ядерным реактором № 4 (фирма EDF, Франция).

Российские ученые и инженеры также решили пойти по этому пути. Минатом РФ приобрел лицензию на производство программируемых контроллеров TELEPERM ME у фирмы "Siemens". Перед разработчиками была поставлена задача спроектировать для АЭС интегрированную АСУТП на их основе. Институту проблем управления им. В.А. Трапезникова РАН было поручено разработать систему верхнего блочного уровня (СВБУ) АСУТП, включая программное обеспечение [1–3].

Система верхнего блочного уровня проектировалась как подсистема АСУТП, задача которой заключается в

интеграции информации и органов управления всех подсистем нормальной эксплуатации АЭС, систем безопасности и специальных систем, к которым относятся подсистемы внутриреакторного контроля, системы автоматического контроля нейтронного потока, система управления компенсирующими стержнями ядерного реактора и другие, всего двенадцать подсистем.

В настоящее время СВБУ, включая техническую и программную составляющие, изготовлена, испытана и поставлена на АЭС.

## 1. ФУНКЦИИ ИНТЕГРИРУЮЩЕГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Функции программного обеспечения (ПО) делятся на информационные, управляющие и вспомогательные.

*Информационные функции* предназначены для передачи информации от программируемых контроллеров до экранов дисплеев, посредством которых оперативный персонал должен контролировать объект. Таких дисплеев много (несколько десятков), персонал насчитывает несколько человек, у каждого своя зона ответственности, которые частично пересекаются (на 10–25 %). Поэтому логика распределения информации по экранам дисплеев имеет весьма сложный характер.

*Функции управления* служат для передачи управляющих воздействий, вводимых с помощью компьютерных органов управления (трекбола, мыши, клавиатуры) в программируемые контроллеры, которые затем осуществляют управление механизмами.

*Вспомогательные функции* обеспечивают требуемые показатели надежности и безопасности эксплуатации АСУТП. К ним, в частности, относятся: ведение единого времени, ведение долговременного архива, контроль отказов элементов АСУТП, обеспечение защиты от не-

санкционированного доступа, управление техническими и программными средствами АСУТП и др.

Рассмотрим конкретный пример.

В СВБУ информационные задачи должны обеспечивать оператора-технолога реакторного отделения оперативной (с задержкой не более двух секунд) информацией о состоянии систем нормальной эксплуатации (около 1500 датчиков, более 1000 механизмов), систем технологической безопасности (около 1000 датчиков, более 500 механизмов), системы внутриреакторного контроля (около 1500 измеряемых и расчетных параметров) и других подсистем (более 3000 аналоговых и дискретных параметров). При этом должна обеспечиваться возможность оперативного (с задержкой не более двух секунд) управления более чем тысячей механизмов, а вспомогательные задачи должны обеспечивать погрешность привязки событий не хуже 10 мс, запись всех событий в долговременный архив, контроль состояния более тысячи контроллеров, защиту их целостности от несанкционированного доступа, надежное управление расчетными задачами.

Общий вывод: интегрирующее ПО, предназначенное для применения в СВБУ, должно выполнять все функции, прямо или косвенно связанные с контролем и управлением, многочисленные вспомогательные функции, а также обеспечивать эффективную работу с большим числом сигналов ( $10^4$ – $10^5$ ), причем временная задержка при передаче информации не должна превышать 2 с.

## 2. ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Сложные интегрированные АСУТП, как правило, управляют большими, энергонасыщенными и потенциально опасными для человека и окружающей среды объектами. Поэтому ПО, работающее в их составе, само является источником опасности.

По нашему мнению, ошибки в ПО могут иметь не менее серьезные последствия, чем дефекты строительных конструкций, технологического и электрического оборудования.

Применительно к АЭС правила разработки и обращения с ПО регламентированы в нормативных документах МАГАТЭ и МЭК, в которых содержатся строгие требования к контролю процесса изготовления ПО на всех этапах его жизненного цикла. Хотя в России эти требования не действуют, при поставках ПО на зарубежные АЭС оно проходит контроль на соответствие международным требованиям по качеству и безопасности.

Это, в частности, относится и к входящему в состав СВБУ программному обеспечению, которое изначально планировалось как изделие на экспорт.

Рамки настоящей статьи не позволяют подробно остановиться на всех этапах процесса разработки ПО, однако стоит выделить несколько моментов.

■ Термин “разработка ПО” в применении к ПО для АСУТП должен относиться ко всем его компонентам, включая операционную систему, драйверы устройств и т. д., а не только к той части, которая непосредственно решает задачи контроля и управления. Это, безусловно, не означает, что все ПО нужно

“переписывать с нуля”. Можно и нужно использовать уже готовое ПО, которое в обязательном порядке следует подвергать процедурам верификации исходного кода и испытаниям по специально разработанным методикам [4].

В полезности и надежности такой строгой проверки убеждает следующий пример. В ПО для СВБУ применяется операционная система, в которую входят компоненты свободно распространяемой операционной системы Linux. В результате верификации исходного кода нами было выяснено, что ядро Linux не может работать более одного года без перезагрузки. Поэтому в регламент эксплуатации СВБУ была введена плановая перезагрузка операционной системы, которая не приводит к отказу СВБУ и не нарушает работу оперативного персонала.

■ Программное обеспечение должно работать в течение всего срока эксплуатации АСУТП, которая может продолжаться до 30–50 лет. В течение такого длительного периода времени многократно изменяются элементная база компьютеров, разрядность и другие параметры, а ПО должно продолжать работать без остановок. Как этого достичь?

Решение состоит в применении консервативных языков программирования, проверенных протоколов передачи информации, которые существуют уже более 20 лет; методов, которые не зависят от особенностей технических средств и носителей информации. В частности, ПО для СВБУ создано на основе таких проверенных временем стандартов как POSIX 1, TCP/IP, X Window и др.

■ По нашему мнению, испытания ПО и верификацию документации наряду с разработчиками должны проводить независимые группы специалистов, которые не участвовали в разработке ПО. Опыт показывает, что ошибки, выявленные независимыми специалистами, отличаются от тех, которые выявляют сами разработчики.

■ Насколько “открытым” должно быть ПО? Под этим полусленговым термином мы понимаем возможность внесения изменений в ПО, а не защиту от взлома. Наш опыт показывает: ПО должно быть открытым на 100 %, т. е. оно должно быть построено и документировано таким образом, чтобы можно было вносить изменения в любой его компонент.

Этот вывод сделан на основе наблюдения за тем способом планирования работ, который практикуется в российских организациях. Он характеризуется, с одной стороны, слабой проработкой технических решений на этапе проектирования, а с другой — низким уровнем исполнительской дисциплины. Применительно к ПО это означает, что перед началом работ, в которых задействованы различные организации и коллективы, не производится полной проработки интерфейсов между компонентами ПО, а те соглашения, которые все же установлены, часто не соблюдаются. Поэтому интегрирующее ПО, в силу своей специфики, должно прирабатываться к различным протоколам, интерфейсам, алгоритмам обработки информации, т. е. быть гибким.

Проиллюстрируем это примером.

Система верхнего блочного уровня для АСУТП должна работать с контроллерами, производимыми по лицензии фирмы “Siemens” Всероссийским научно-иссле-



довательским институтом измерений и автоматики. Они обладают следующей особенностью: наряду с сигналами о состоянии механизмов, которые формируются инициативно и передаются в СВБУ, в них используются так называемые запросные сигналы, для получения которых в СВБУ необходимо формировать специальные запросы к контроллерам. Число таких контроллеров достигает более одной тысячи, а конструкция сети CS275 фирмы "Siemens" не позволяет производить более ста запросов в секунду без перегрузки сетевого трафика. Поэтому оперативное (с задержкой не более двух секунд) получение значений этих запросных сигналов невозможно простым циклическим опросом контроллеров.

Для решения проблемы оперативного представления значений запросных сигналов в состав интегрирующего ПО пришлось ввести специальный интеллектуальный алгоритм, который анализирует, какие именно запросные сигналы необходимы для управления в текущий момент времени.

### 3. СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ СВБУ

Остановимся на особенностях ПО для СВБУ, которое является примером интегрирующего ПО для АСУТП.

В качестве базового языка программирования был выбран язык программирования Си стандарта POSIX 1. Это один из наиболее распространенных языков программирования среднего уровня, который существует уже десятилетия и будет существовать еще десятки лет.

Непосредственно на этом языке реализовано так называемое интерфейсное программное обеспечение, которое представляет собой библиотеку программ и применяется для создания "шлюзовых" программ, предназначенных для обеспечения взаимодействия с программируемыми контроллерами и расчетными программами.

На основе языка Си и применения графической библиотеки Motif разработана программа, обеспечивающая отображение информации в графической и текстовой форме на любой UNIX или Linux машине, X-терминале или эмуляторе X-терминала под Windows.

Однако для реализации основных алгоритмов обработки информации используется язык логического программирования высокого уровня ABIS — оригинальная разработка Института проблем управления (ИПУ) [5, 6]. Он реализован также на основе языка Си, работает практически на всех операционных системах и может быть портирован на любую операционную систему. Язык ABIS чрезвычайно компактен и позволяет просто и быстро реализовывать сложные распределенные программные комплексы. Так, в частности, объем исходного ABIS-кода ПО для СВБУ не превышает 500 килобайт. При этом язык достаточно эффективен для задач АСУТП. Секрет его успешного применения состоит в том, что в нем реализованы действенные алгоритмы работы с базами данных. Это позволило использовать в качестве технических средств СВБУ слабые процессоры с тактовой частотой 500—760 МГц для обработки больших объемов информации с приемлемой скоростью (см. § 1).

В качестве операционной системы, которая рассматривается как часть интегрирующего ПО, используется оригинальная разработка ИПУ, называемая LICS (Linux

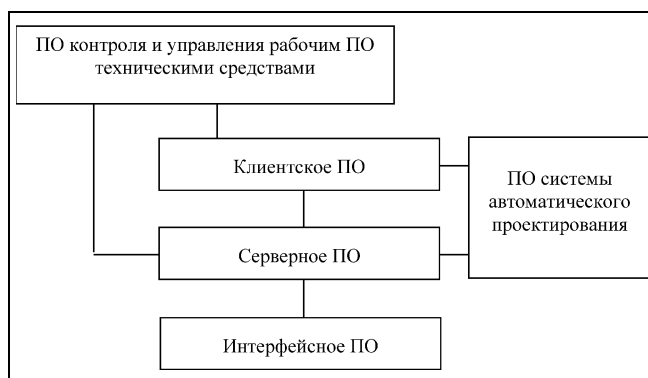


Рис. 1. Структура программного обеспечения

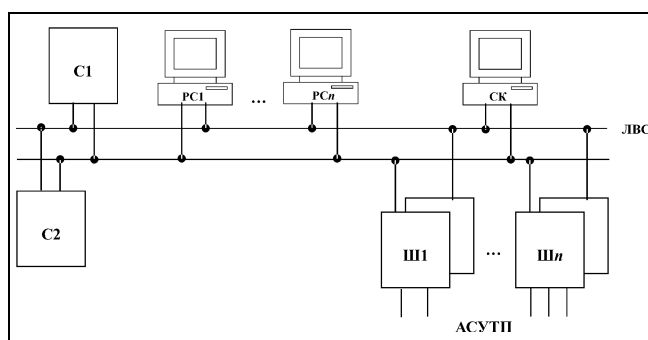


Рис. 2. Структура технических средств

of the Institute of Control Science) [7] (см. третью страницу обложки). Она построена на основе свободно распространяемого ПО и является представителем семейства Linux, но имеет некоторые преимущества по отношению к другим представителям этого семейства. Все компоненты операционной системы LICS прошли верификацию и испытания. Данная операционная система обладает лицензионной чистотой и независимостью от ограничений каких бы то ни было государств, фирм и организаций.

На основе перечисленных базовых компонентов был разработан комплекс программ для разработки СВБУ, представленный на рис. 1.

Помимо интерфейсного ПО, в состав комплекса входят:

- серверное ПО, которое размещается на серверах СВБУ и осуществляет основные алгоритмы обработки и сохранения информации;
- клиентское ПО, осуществляющее диалог с операторами-технологами;
- ПО системы автоматического проектирования;
- специализированное ПО, в функции которого входит обеспечение централизованного дистанционного контроля всех компонентов ПО и технических средств, на которых оно функционирует.

Типовая схема технических средств, на которых инсталлируется данное ПО, представлена на рис. 2, где  $C1$  и  $C2$  — основной и резервный серверы, на которых инс-

таллировано серверное ПО; РС1, ..., РС<sub>n</sub> — рабочие станции с одним или несколькими дисплеями, на которых инсталлировано клиентское ПО; Ш1, ..., Ш<sub>n</sub> — шлюзовые компьютеры, на которых инсталлировано интерфейсное ПО; СК — рабочая станция, на которой инсталлировано ПО контроля и управления рабочим ПО и технических средств; ЛВС — дублированная локальная вычислительная сеть.

### ЗАКЛЮЧЕНИЕ

Из приведенных рассуждений можно сделать вывод, что интегрирующее ПО должно обладать следующими основными свойствами:

- выполнять все необходимые потребительские функции с заявленными временными, надежностными и прочими параметрами;
- быть спроектированным, изготовленным и протестированным по правилам, установленным в нормативной документации;
- быть максимально гибким и открытым.

Автор надеется, что приведенные в этой работе сведения будут полезны при разработке современных интегрирующих программно-технических комплексов в различных областях науки и техники.

Более подробную информацию о СВБУ и компонентах ПО можно найти на сайте [www.31.ipu.rssi.ru](http://www.31.ipu.rssi.ru).

### ЛИТЕРАТУРА

1. Основные решения по созданию системы верхнего (блочного) уровня АСУТП АЭС / А. Г. Полетыкин, М. Е. Бывайков, Н. Э. Менгазетдинов, А. А. Байбулатов // Ядерные измерительно-информационные технологии. — 2004. — № 1—2.
2. Основные решения по созданию системы верхнего (блочного) уровня АСУТП АЭС / А. Г. Полетыкин, М. Е. Бывайков, Н. Э. Менгазетдинов, А. А. Байбулатов // Труды Ин-та проблем управления. — 2002. — Т. XVIII.
3. Опыт проектирования системы верхнего (блочного) уровня АСУТП АЭС / А. Г. Полетыкин, Н. Э. Менгазетдинов, М. Е. Бывайков и др. // Вторая междунар. конф. по проблемам управления / Ин-т проблем управления. — М., 2003.
4. Некоторые аспекты применения свободно распространяемых программных продуктов в АСУТП АЭС / А. В. Антонов, А. А. Байбулатов, С. И. Масолкин и др. // Труды Ин-та проблем управления. — 2001. Т. XIII.
5. Полетыкин А. Г., Байбулатов А. А. Основы языка ABIS // II междунар. конф. "Идентификация систем и задачи управления" / Ин-т проблем управления. — М., 2003.
6. Полетыкин А. Г., Бывайков М. Е., Байбулатов А. А. Язык логического программирования — ABIS // Труды Ин-та проблем управления. — 2004. — Т. XXIV.
7. Системное программное обеспечение LICS как компонент подсистем АСУТП АЭС / С. И. Масолкин, В. Г. Промыслов, Е. Ф. Жарко и др. // Автоматизация в промышленности. — 2004. — № 10.

☎ (095) 334-75-71



## XIII МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ

### "ПРОБЛЕМЫ УПРАВЛЕНИЯ БЕЗОПАСНОСТЬЮ СЛОЖНЫХ СИСТЕМ"

(Москва, декабрь 2005 г.)

**Предполагается рассмотреть:** проблемы и методы оценки безопасности различного типа; механизмы управления безопасностью; правовое регулирование вопросов безопасности; формирование структур систем управления безопасностью; теорию и методы принятия решений, связанные с безопасностью; прогнозирование и моделирование процессов управления безопасностью; планирование и стратегическое управление в системах обеспечения безопасности; методы построения средств информационной поддержки принятия решений в системах управления безопасностью; системы управления силами и средствами при управлении безопасностью.

**Конференция состоится** в Институте проблем управления им. В.А. Трапезникова РАН по адресу: Москва, Профсоюзная ул., 65. Официальные языки конференции — русский, английский. Продолжительность работы конференции — 1 день.

**Заявки** на участие в конференции принимаются по адресу: 117997 Москва, ГСП-7, Профсоюзная ул., 65, Институт проблем управления, лаб. 20, Оргкомитет международной конференции; тел. (095) 334-89-59, e-mail: [Conf20@ipu.rssi.ru](mailto:Conf20@ipu.rssi.ru)

**Материалы** представляются на дискете плюс 1 экз. в распечатанном виде (2–4 стр.). На этикетке дискеты указать ф.и.о. авторов и имя файла, названного по фамилии первого автора; или высылаются по электронной почте. В графе "Тема" укажите — Конференция. Необходимо сообщить сведения об авторах: фамилию, имя, отчество автора(ов); должность, ученое звание; место работы (полное название и аббревиатура); почтовый адрес для переписки (обязательно указать индекс) и(или) e-mail; номер телефона для связи.

**Требования по оформлению.** Материалы должны быть представлены в редакторе Word, версии не ниже 6.0; формат А4, заполняемый текстом 115×165 мм (параметры страницы: верхнее поле — 2,5 см; нижнее — 10,7 см; левое — 4,7 см; правое — 4,8 см). Шрифт — Times New Roman, 10 пунктов через 1 интервал, красная строка — 0,5 см, страницы не нумеруются. Библиографические ссылки в тексте даются в квадратных скобках, рисунки должны допускать возможность масштабирования.



# ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ РАЗРЫВНЫХ ФУНКЦИЙ

В. И. Левин

*Пензенская государственная технологическая академия*

Показана возможность аналитического моделирования разрывных функций с помощью операций непрерывной логики. Отмечено, что линейный относительно числа разрывов функции рост трудоемкости моделирования позволяет синтезировать разрывные кривые в виде логической суперпозиции простых непрерывных кривых.

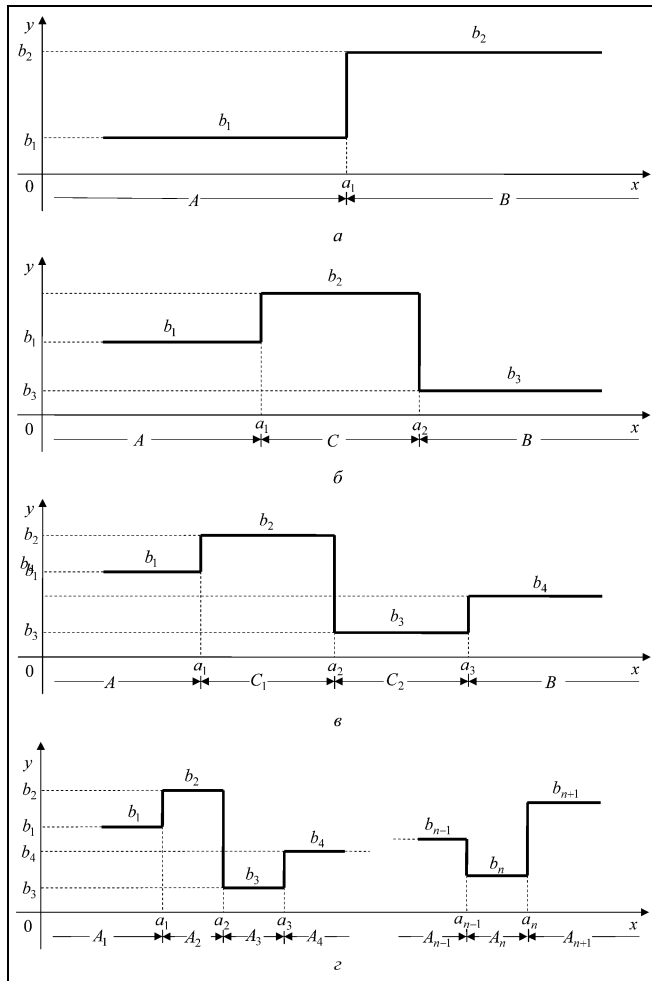
## ВВЕДЕНИЕ

Последние 50 лет большое внимание исследователей привлекает так называемая непрерывная логика (НЛ). Сфера применения НЛ очень широка, она охватывает технические, экономические, социальные и иные системы. С помощью НЛ решаются весьма разнородные задачи: аппроксимация функций, синтез функциональных генераторов и преобразователей формы информации, расчет нелинейных цепей, изучение динамики автоматов и их диагностика, анализ надежности систем, распознавание образов и сцен, построение теории нечетких множеств и принятие решений в условиях неопределенности, обработка информации, анализ систем обслуживания, дискретная оптимизация и теория расписаний, математическое моделирование социальных, экономических и исторических процессов и др. [1–3]. Особое значение для техники имеет возможность решения с помощью НЛ обратных задач аналитической геометрии, т. е. аналитическое представление с помощью соответствующих уравнений НЛ заданных кривых, поверхностей и объемов. Применение в этой области аппарата НЛ позволяет описывать в аналитической форме нелинейные, ломаные и многозначные кривые и поверхности, а также объемы и их границы. Это дает возможность аналитического представления различных характеристик систем управления и других систем, с последующим их формализованным синтезом. Однако для одного, практически очень важного класса задач, аппарат НЛ до последнего времени не применялся, а именно, для аналитического представления разрывных кривых и поверхностей. Многие исследователи и сегодня считают, что для решения указанного класса задач аппарат НЛ в принципе не применим. Эта точка зрения привела к разработке альтернативных математических аппаратов, направленных на решение указанных задач [4].

В настоящей статье показано, что НЛ вполне применима к решению задач аналитического моделирования разрывных кривых, в виде которых часто представляются характеристики систем управления и других технических систем. При этом аппарат НЛ оказывается совершенно естественным средством решения указанных задач.

## 1. ПОСТАНОВКА ЗАДАЧИ

Пусть имеется некоторая разрывная кривая с одним, двумя или несколькими разрывами первого рода, в которой участки между соседними точками разрыва образованы отрезками горизонтальных прямых (см. рисунок). Рассмотрение только разрывных кривых простейшего вида, показанных на рисунке (стандартных разрывных кривых), не ограничивает общности, поскольку хорошо известно, что произвольные разрывные кривые с разрывами первого рода всегда можно выразить через стандартные кривые с такими разрывами [1, 2]. Например, разрывную кривую  $f(x)$ , описываемую при  $x < a$  выражением  $f_1(x)$ , а при  $x > a$  — выражением  $f_2(x)$  (т. е. терпящую в точке  $a$  разрыв первого рода), аналитически можно представить в виде  $f(x) = f_1(x)I(a - x) + f_2(x)I(x - a)$ , где  $I(a)$  — стандартная разрывная кривая простейшего вида, определяемая как  $I(x) = \begin{cases} 1 & \text{при } x > 0 \\ 0 & \text{при } x < 0 \end{cases}$  (единичная функция). Таким образом, нашу задачу, без ограничения общности, можно сформулировать как задачу аналитического моделирования стандартных разрывных кривых, типа показанных на рисунке, с использованием операций НЛ. Эти операции вводятся следующим образом [1–3]. Пусть  $C = [A, B]$  — отрезок на оси вещественных чисел, с серединой в точке  $M = (A + B)/2$ .



### Простейшие разрывные кривые:

*a* — с двумя участками непрерывности и одним разрывом;  
*б* — с тремя участками непрерывности и двумя разрывами;  
*в* — с четырьмя участками непрерывности и тремя разрывами;  
*г* — с  $(n + 1)$  участками непрерывности и  $n$  разрывами

Тогда базовые операции НЛ — дизъюнкция, конъюнкция и отрицание — вводятся в виде

$$a \vee b = \max(a, b), \quad a \wedge b = \min(a, b), \\ \bar{a} = 2M - a, \quad a, b \in C. \quad (1)$$

Другие возможные операции НЛ вводятся через базовые операции (1). Это: включение  $a \supset b = (B - a + b) \wedge B$ ; импликация  $a \rightarrow b = \bar{a} \vee b$ ; эквивалентность  $a \sim b = (a \vee \bar{b}) \wedge (\bar{a} \vee b)$ ; неэквивалентность (исключающая дизъюнкция)  $a \nabla b = (a \wedge \bar{b}) \vee (\bar{a} \wedge b)$ ; операция Шеффера  $a \downarrow b = \overline{a \wedge b}$ ; операция Вебба  $a \uparrow b = \overline{a \vee b}$ ; противоречие  $a \neq b = a \wedge \bar{a}$ ; тавтология  $(a = b) = a \vee \bar{a}$ ; запрет  $a \rightarrow b = a \wedge \bar{b}$ .

Поскольку все введенные операции НЛ выражаются через базовые операции НЛ (1), можно ожидать, что решение поставленной задачи, если оно существует, может быть получено в терминах только базовых операций НЛ — дизъюнкции, конъюнкции и отрицания.

## 2. ИДЕЯ РЕШЕНИЯ

Идея решения поставленной задачи проста и заключается в следующем. Рассмотрим стандартную разрывную кривую простейшего вида (см. рис. *a*) с двумя участками непрерывности и одним разрывом между ними. Ее уравнение можно записать в виде совокупности двух уравнений, соответствующих двум участкам кривой

$$f_A(x, y) = 0 \text{ на участке } A; \\ f_B(x, y) = 0 \text{ на участке } B, \quad (2)$$

где  $f_A(x, y)$  и  $f_B(x, y)$  — функции, соответствующие участкам *A* и *B* кривой.

В общем случае произвольная точка  $M = (x, y)$  на рис. *a* может находиться: 1) на участке *A* кривой; 2) на участке *B* кривой; 3) вне кривой. Между этими тремя возможными положениями точки *M* и соответствующими значениями функций  $f_A(x, y)$  и  $f_B(x, y)$  имеется следующее взаимнооднозначное соответствие:

$$\begin{aligned} \text{случай 1: } f_A(x, y) = 0, f_B(x, y) \neq 0; \\ \text{случай 2: } f_A(x, y) \neq 0, f_B(x, y) = 0; \\ \text{случай 3: } f_A(x, y) \neq 0, f_B(x, y) \neq 0. \end{aligned} \quad (3)$$

Предположим, что функции  $f_A(x, y)$  и  $f_B(x, y)$ , с помощью которых записываются уравнения (2) участков *A* и *B* кривой (см. рис. *a*), удовлетворяют более определенным, чем выше, условиям:

$$f_A(x, y) > 0 \text{ вне участка } A; \\ f_B(x, y) > 0 \text{ вне участка } B. \quad (4)$$

Условия (4) означают, что указанные функции, которые, будучи приравненными нулю, описывают — каждая на своем участке (см. уравнения (2)) — кривую на рис. *a*, в любых других точках рис. *a*, находящихся вне этого участка, имеют положительное значение. С учетом условий (4) соответствие (3) принимает вид:

$$\begin{aligned} \text{случай 1: } f_A(x, y) = 0, f_B(x, y) > 0; \\ \text{случай 2: } f_A(x, y) > 0, f_B(x, y) = 0; \\ \text{случай 3: } f_A(x, y) > 0 \text{ или } f_A(x, y) < 0, f_B(x, y) > 0 \\ \text{или } f_B(x, y) < 0. \end{aligned} \quad (5)$$

Из выражений (5) видно, что функция  $f(x, y)$ , обращающаяся в нуль на кривой (см. рис. *a*) и не равная нулю вне этой кривой, имеет вид

$$f(x, y) = f_A(x, y) \wedge f_B(x, y), \quad (6)$$

где  $\wedge$  — операция конъюнкции НЛ. Таким образом, уравнение кривой записывается с помощью введенной функции  $f(x, y)$  как

$$f(x, y) = 0. \quad (7)$$

После подстановки в уравнение (7) выражения этой функции (6) получим явную форму искомого уравнения кривой:

$$f_A(x, y) \wedge f_B(x, y) = 0, \quad (8)$$

выраженную через уравнения (2) обоих участков этой кривой. Как видно из уравнения (8), если функции  $f_A(x, y)$  и



$f_B(x, y)$  с помощью которых записываются уравнения (2) участков  $A$  и  $B$  разрывной кривой, удовлетворяют условиям (4), эти частные уравнения можно “сшить” в единое уравнение (8) кривой с помощью операции конъюнкции НЛ. Таким образом, для получения уравнения разрывной кривой (см. рис. *a*) надо представить участки  $A$  и  $B$  такими уравнениями (2), в которых представляющие функции  $f_A$  и  $f_B$  удовлетворяют условию (4).

Описанный “метод сшивания” пригоден для получения не только уравнения простейшей разрывной кривой с двумя участками, но и уравнений любых стандартных разрывных кривых. Так, для кривой (см. рис. *b*) с тремя участками непрерывности и двумя разрывами между ними описывающее ее уравнение можно представить по участкам в виде совокупности трех уравнений, соответствующих трем участкам кривой:

$$f_A(x, y) = 0 \text{ на участке } A; f_B(x, y) = 0 \text{ на участке } B; f_C(x, y) = 0 \text{ на участке } C, \quad (9)$$

где  $f_A(x, y)$ ,  $f_B(x, y)$  и  $f_C(x, y)$  — функции, соответствующие участкам  $A$ ,  $B$  и  $C$  кривой. Если эти функции удовлетворяют условиям типа (4)

$$\begin{aligned} f_A(x, y) &> 0 \text{ вне участка } A; \\ f_B(x, y) &> 0 \text{ вне участка } B; \\ f_C(x, y) &> 0 \text{ вне участка } C, \end{aligned} \quad (10)$$

то имеется следующее взаимнооднозначное соответствие между различными возможными положениями произвольной точки  $M(x, y)$  на рис. *b* и значениями функций  $f_A$ ,  $f_B$  и  $f_C$ :

- случай 1:  $M$  находится на участке  $A$  кривой,  
 $f_A(x, y) = 0, f_B(x, y) > 0, f_C(x, y) > 0$ ;
- случай 2:  $M$  находится на участке  $B$  кривой,  
 $f_B(x, y) = 0, f_A(x, y) > 0, f_C(x, y) > 0$ ; (11)
- случай 3:  $M$  находится на участке  $C$  кривой,  
 $f_C(x, y) = 0, f_A(x, y) > 0, f_B(x, y) > 0$ ;
- случай 4:  $M$  находится вне кривой,  $f_A(x, y) > 0$   
или  $f_B(x, y) < 0$  или  $f_C(x, y) < 0$   
или  $f_C(x, y) < 0$ .

Как видно из выражений (11), функция  $f(x, y)$ , равная нулю на кривой (см. рис. *b*) и отличная от нуля вне этой кривой, имеет вид:  $f(x, y) = f_A(x, y) \wedge f_B(x, y) \wedge f_C(x, y)$ .

Таким образом, если функции  $f_A$ ,  $f_B$  и  $f_C$ , которые описывают уравнения (9) участков  $A$ ,  $B$  и  $C$  кривой рис. *b*, удовлетворяют условиям (10), то явная форма уравнения кривой в терминах операций НЛ такова:

$$f_A(x, y) \wedge f_B(x, y) \wedge f_C(x, y) = 0. \quad (12)$$

Таким образом, и в случае разрывной кривой (см. рис. *b*) с тремя участками непрерывности, если функции  $f_A(x, y)$ ,  $f_B(x, y)$  и  $f_C(x, y)$ , с помощью которых записываются уравнения (9) участков  $A$ ,  $B$  и  $C$  кривой, удовлетворяют условиям (10), эти частные уравнения “сшиваются” в общее уравнение (12) кривой с помощью операции конъюнкции НЛ.

Аналогично, уравнение разрывной кривой (см. рис. *в*) с четырьмя участками непрерывности (и тремя разрывами между ними), которые описываются уравнениями

$$\begin{aligned} f_A(x, y) &= 0 \text{ на участке } A; f_B(x, y) = 0 \text{ на участке } B; \\ f_{C_1}(x, y) &= 0 \text{ на участке } C_1; \\ f_{C_2}(x, y) &= 0 \text{ на участке } C_2 \end{aligned} \quad (13)$$

с функциями  $f(\cdot)$ , удовлетворяющими условиям

$$\begin{aligned} f_A(x, y) &> 0 \text{ вне участка } A; f_B(x, y) > 0 \text{ вне участка } B; \\ f_{C_1}(x, y) &> 0 \text{ вне участка } C_1; \\ f_{C_2}(x, y) &> 0 \text{ вне участка } C_2, \end{aligned} \quad (14)$$

записываются с помощью операций НЛ в виде  $f_A(x, y) \wedge f_B(x, y) \wedge f_{C_1}(x, y) \wedge f_{C_2}(x, y) = 0$ .

Вообще, если имеется разрывная кривая общего вида (см. рис. *г*) с произвольным числом  $n + 1$  участков  $A_1, \dots, A_{n+1}$  (и  $n$  разрывами между ними), описываемых уравнениями

$$f_A(x, y) = 0, \quad A \in \{A_1, \dots, A_{n+1}\}, \quad (15)$$

с функциями  $f(\cdot)$ , удовлетворяющими условиям

$$f_A(x, y) > 0 \text{ вне участка } A, \quad A \in \{A_1, \dots, A_{n+1}\}, \quad (16)$$

то уравнение этой кривой можно записать с помощью операций НЛ в виде

$$\bigwedge_{A \in \{A_1, \dots, A_{n+1}\}} f_A(x, y) = 0.$$

Из изложенного следует, что поставленная задача аналитического моделирования стандартных разрывных кривых (типа показанных на рисунке) с помощью НЛ сводится к гораздо более простой задаче аналитического моделирования отдельных непрерывных участков таких кривых, имеющих вид отрезков горизонтальных прямых, с помощью той же НЛ. При этом функции, с помощью которых описываются указанные участки (см. уравнения (2), (9), (13) и (15)), должны быть положительными на всех других участках (см. условия (4), (10), (14) и (16)), что, как мы увидим далее, легко достижимо.

### 3. МОДЕЛИРОВАНИЕ УЧАСТКОВ СТАНДАРТНЫХ РАЗРЫВНЫХ КРИВЫХ

Как следует из рисунка, имеются всего три типа горизонтальных непрерывных участков стандартных разрывных кривых. Участок первого типа представляет собой ограниченный слева и неограниченный справа горизонтальный отрезок  $B$ , задаваемый в виде системы уравнений и неравенств

$$\left. \begin{aligned} y &= b \\ x &> a \end{aligned} \right\}. \quad (17)$$

Перепишем эту систему в виде

$$\left. \begin{aligned} y - b &= 0 \\ a - x &< 0 \end{aligned} \right\}. \quad (18)$$

В терминах операции НЛ — дизъюнкции — систему (18) можно представить как одно уравнение

$$f_B(x, y) \equiv (y - b) \vee (a - x) = 0. \quad (19)$$

Итак, любая точка  $(x, y)$  на изучаемом отрезке (17) удовлетворяет уравнению (19). С другой стороны, любая точка  $(x, y)$ , находящаяся вне отрезка (17), удовлетворяет хотя бы одному из условий:  $y > b$ ,  $y < b$ ,  $x < a$ , вследствие чего для нее не выполняется уравнение (19). Таким образом, уравнение (19) является уравнением ограниченного слева и неограниченного справа горизонтального отрезка типа (17), записанным с помощью операции дизъюнкции НЛ. При этом за пределами отрезка, т. е. при  $x < a$ , левая часть уравнения (19) положительна, т. е. выполнено условие (16).

Горизонтальный участок второго типа представляет собой ограниченный справа и неограниченный слева горизонтальный отрезок  $A$ , задаваемый системой уравнений и неравенств

$$\left. \begin{array}{l} y = b \\ x < a \end{array} \right\}. \quad (20)$$

Переписав систему (20) в виде

$$\left. \begin{array}{l} y - b = 0 \\ x - a < 0 \end{array} \right\},$$

аналогично представляем ее единым логическим уравнением

$$f_A(x, y) \equiv (y - b) \vee (a - x) = 0 \quad (21)$$

и убеждаемся, что любая точка  $(x, y)$  на изучаемом отрезке (20) удовлетворяет уравнению (21), а любая точка вне этого отрезка не удовлетворяет ему, т. е. уравнение (21) есть уравнение ограниченного справа и неограниченного слева горизонтального отрезка типа (20), записанное с помощью операции дизъюнкции НЛ. Причем за пределами отрезка, т. е. при  $x > a$ , левая часть уравнения (21) положительна, т. е. выполнено условие (16).

Наконец, горизонтальный участок третьего типа является ограниченным слева и справа горизонтальным отрезком  $C$ , который можно задать в виде системы уравнений и неравенств

$$\left. \begin{array}{l} y = b \\ x > a \\ x < c \end{array} \right\}, \quad a < c. \quad (22)$$

Систему (22) можно переписать в виде

$$\left. \begin{array}{l} y - b = 0 \\ a - x < 0 \\ x - c < 0 \end{array} \right\}, \quad a < c. \quad (23)$$

Системе (23) соответствует единое логическое уравнение вида

$$f_C(x, y) \equiv (y - b) \vee (a - x) \vee (x - c) = 0. \quad (24)$$

Убеждаемся, что любая точка  $(x, y)$  на отрезке (22) удовлетворяет уравнению (24), а любая точка вне этого отрезка не удовлетворяет ему. Поэтому уравнение (24) и

есть нужное нам уравнение ограниченного слева и справа горизонтального отрезка типа (22), записанное с помощью операции дизъюнкции НЛ. При этом опять за пределами отрезка, т. е. при  $(x < a) \cup (x > c)$ , левая часть уравнения (24) положительна, т. е. выполнено условие (16).

Итак, все три возможных типа горизонтальных непрерывных участков стандартных разрывных кривых, показанных на рисунке, можно аналитически моделировать соответствующими логическими уравнениями (19), (21) и (24) со специальной (удовлетворяющей дополнительно свойству (16)) левой частью.

#### 4. ВЫВОД УРАВНЕНИЙ СТАНДАРТНЫХ РАЗРЫВНЫХ КРИВЫХ

Как показано в § 3, любой горизонтальный непрерывный участок  $A$  стандартной разрывной кривой, типа представленных на рисунке, можно промоделировать аналитически в виде уравнения (15) с помощью некоторой функции  $f_A(x, y)$ , сформированной с помощью операций вычитания и непрерывно-логической дизъюнкции и удовлетворяющей условию (16). Это открывает возможность общего логико-аналитического моделирования указанных разрывных кривых путем “сшивания” уравнений их отдельных непрерывных горизонтальных участков в единое уравнение всей разрывной кривой (см. § 2).

Начнем с вывода логического уравнения простейшей стандартной разрывной кривой с двумя участками непрерывности и одним разрывом между ними (см. рис. *a*). Согласно § 2, общий вид уравнения этой кривой есть уравнение (8), при условии, что уравнения первого ( $A$ ) и второго ( $B$ ) ее участков имеют вид (2) и входящие в эти уравнения функции — левые части  $f_A(x, y)$  и  $f_B(x, y)$  — удовлетворяют условиям (4). Но, согласно § 3, уравнение участка  $A$  есть уравнение (21) при  $b = b_1$ ,  $a = a_1$ , уравнение участка  $B$  — это уравнение (19) при  $b = b_2$ ,  $a = a_1$ . При этом оба уравнения имеют вид (2), а их левые части  $f_A(x, y)$  и  $f_B(x, y)$  удовлетворяют условиям (4). Следовательно, уравнение кривой (см. рис. *a*) имеет вид (8). Подставив в уравнение (8) конкретизированные согласно указанному левые части уравнений участков  $A$  и  $B$  — функции  $f_A(x, y)$  и  $f_B(x, y)$ , получим искомое уравнение кривой с двумя участками непрерывности и одним разрывом между ними в форме

$$[(y - b_1) \vee (x - a_1)] \wedge [(y - b_2) \vee (a_1 - x)] = 0. \quad (25)$$

Как видно, наше уравнение содержит в левой части четыре алгебраические операции вычитания, две логические операции дизъюнкции НЛ и одну логическую операцию конъюнкции НЛ, т. е. две операции вычитания и одну операцию дизъюнкции на один участок и одну операцию конъюнкции на один разрыв кривой.

Воспользуемся тем же приемом для вывода логического уравнения следующей по сложности стандартной разрывной кривой, имеющей три участка непрерывности и два разрыва между ними (см. рис. *b*). Общий вид уравнения этой кривой есть (12), при условии, что уравнения первого ( $A$ ), второго ( $C$ ) и третьего ( $B$ ) ее участков имеют вид (9); входящие в эти уравнения функции — левые части  $f_A(x, y)$ ,  $f_B(x, y)$  и  $f_C(x, y)$  — удовлетворяют





условиям (10). Согласно § 3, уравнение участка  $A$  есть уравнение (21) при  $b = b_1$ ,  $a = a_1$ , уравнение участка  $C$  — уравнение (24) при  $b = b_2$ ,  $a = a_1$ ,  $c = a_2$ , уравнение участка  $B$  — уравнение (19) при  $b = b_3$ ,  $a = a_2$ . При этом все три уравнения имеют вид (9), а их левые части  $f_A(x, y)$ ,  $f_B(x, y)$  и  $f_C(x, y)$  удовлетворяют условиям (10). Следовательно, уравнение кривой (см. рис. б) имеет вид (12). Подставив в уравнение (12) конкретизированные, как сказано ранее, левые части уравнений участков  $A$ ,  $B$  и  $C$ , т. е. функции  $f_A(x, y)$ ,  $f_B(x, y)$  и  $f_C(x, y)$ , получим нужное уравнение кривой с тремя участками непрерывности и двумя разрывами между ними в форме

$$[(y - b_1) \vee (x - a_1)] \wedge [(y - b_2) \vee (a_1 - x) \vee (x - a_2)] \wedge [(y - b_3) \vee (a_2 - x)] = 0. \quad (26)$$

Уравнение содержит семь алгебраических операций вычитания, четыре логические операции дизъюнкции НЛ и две логические операции конъюнкции НЛ. На один участок (разрыв) кривой в ее уравнении приходится: две операции вычитания на каждый из двух крайних участков плюс три такие операции на средний участок; одна операция дизъюнкции на каждый из двух крайних участков плюс две такие операции на средний участок; одна операция конъюнкции на каждый разрыв кривой.

Описанная методика остается в силе и для произвольных кривых этого класса, содержащих произвольное число участков непрерывности и разрывов между ними. Необходимо лишь учитывать, что последовательное усложнение этих кривых проявляется только в увеличении числа их средних участков, при постоянстве числа крайних участков, равного двум. Этим путем, например, для кривой с четырьмя участками непрерывности (два крайних и два средних) и тремя разрывами между ними (см. рис. в) получим уравнение

$$[(y - b_1) \vee (x - a_1)] \wedge [(y - b_2) \vee (a_1 - x) \vee (x - a_2)] \wedge [(y - b_3) \vee (a_2 - x) \vee (x - a_3)] \wedge [(y - b_4) \vee (a_3 - x)] = 0. \quad (27)$$

Оно содержит десять алгебраических операций вычитания, шесть логических операций дизъюнкции НЛ и три логические операции конъюнкции НЛ. Таким образом, на один участок (разрыв) кривой в ее уравнении приходится: две операции вычитания на каждый из двух крайних участков и три операции вычитания на каждый средний участок; одна операция дизъюнкции на каждый из двух крайних участков и две операции дизъюнкции на каждый средний участок; одна операция конъюнкции на каждый разрыв кривой.

В общем случае для произвольной стандартной разрывной кривой с  $n + 1$  участками непрерывности и  $n$  разрывами между ними,  $n \geq 1$  (см. рис. г), уравнение имеет вид:

$$[(y - b_1) \vee (x - a_1)] \wedge \left\{ \bigwedge_{i=1}^{n-1} [(y - b_{i+1}) \vee (a_i - x) \vee (x - a_{i+1})] \right\} \wedge [(y - b_n) \vee (a_n - x)] = 0. \quad (28)$$

На один участок (разрыв) приходится такое же число различных операций, как и для кривой с четырьмя уча-

сками и тремя разрывами между ними (см. пояснения к уравнению (27)). Это позволяет легко сосчитать общее число различных операций, необходимых для логического моделирования произвольной стандартной разрывной кривой с  $n + 1$  участками непрерывности и  $n$  разрывами между ними с помощью уравнения (28). Обозначив через  $N_{\circ}(n)$  число операций типа “О”, а через  $N(n)$  — общее число операций всех типов, получим

$$N_{-}(n) = 2 + 2 + 3(n - 1) = 3n + 1,$$

$$N_{\vee}(n) = 1 + 1 + 2(n - 1) = 2n,$$

$$N_{\wedge}(n) = 1 + 1 + (n - 2) = n,$$

$$N(n) = N_{-}(n) + N_{\vee}(n) + N_{\wedge}(n) = 6n + 1.$$

Как видно из этих оценок, трудоемкость логического моделирования стандартных разрывных кривых уравнениями вида (25)—(28), измеряемая числом операций различного вида в этих уравнениях, а также суммарным числом этих операций, растет линейно относительно числа  $n$  разрывов кривой. Это позволяет моделировать с помощью указанных уравнений кривые практически неограниченной сложности (с практически неограниченным числом разрывов).

## ЗАКЛЮЧЕНИЕ

В настоящей статье показана возможность аналитического моделирования произвольных разрывных кривых, практически неограниченной сложности, с использованием операций непрерывной логики. Таким образом, теперь можно говорить о логическом моделировании разрывных кривых. Это открывает возможность последующего формализованного логического синтеза таких кривых, с использованием хороших конструктивных качеств языка алгебры логики (в данном случае — непрерывной логики), позволяющего, как видно из изложенного, выражать разрывные функции в виде логико-алгебраической суперпозиции непрерывных функций. Последние, как хорошо известно, могут быть легко реализованы как схемным, так и программным путем и, таким образом, могут быть взяты за элементарные функции при указанном логическом синтезе разрывных кривых.

## ЛИТЕРАТУРА

1. Гинзбург С. А. Непрерывная логика и ее применения // Автоматика и телемеханика. — 1967. — № 2. — С. 115—132.
2. Левин В. И. Непрерывная логика и ее применение. I, II // Там же. — 1990. — № 8. — С. 3—22; № 9. — С. 3—26.
3. Левин В. И. Методы непрерывной логики в задачах управления // Там же. — 2003. — № 3. — С. 28—51.
4. Волгин Л. И. Предикатная алгебра выбора и ее модификации // Опыт, результаты, проблемы повышения конкурентоспособности радиоэлектронной аппаратуры. — Таллинн: Валгус, 1986. — Вып. 4. — С. 64—104.

☎ (8412) 49-61-56

E-mail: levin@pti.ac.ru



# ИТЕРАЦИОННЫЙ АЛГОРИТМ ПОСТРОЕНИЯ ЦИКЛОВ НЕЛИНЕЙНЫХ АВТОНОМНЫХ СИСТЕМ.

## Ч. 2. Оценки параметров

И. Г. Исмаилов

Институт проблем управления им. В. А. Трапезникова, г. Москва

Найдены оценки управляющих параметров итерационного алгоритма в терминах правой части системы, которые могут быть полезны для непосредственного численного поиска неустойчивых циклов нелинейных автономных систем.

### ВВЕДЕНИЕ

В работе [1] рассмотрена задача о приближенном построении цикла многоконтурной автономной системы автоматического регулирования, которую можно привести к виду

$$\frac{dx}{dt} = f(x), \quad x \in \mathbb{R}^N. \quad (1)$$

Для определения точки периодического решения и соответствующего периода была предложена следующая процедура:

$$x_{k+1} = x_k - \gamma_k((I - V^T(t_k, x_k))(x_k - p(t_k, x_k)) + \langle a, x_k \rangle - b)a, \quad (2)$$

$$t_{k+1} = t_k + \mu_k \langle f(p(t_k, x_k)), x_k - p(t_k, x_k) \rangle, \quad k = 0, 1, 2, \dots, \quad (3)$$

где  $V(t, x)$ ,  $V(0, x) = I$ , — фундаментальная матрица линейной системы дифференциальных уравнений  $dV/dt = f'_x(p(t, x))V$ ;  $\gamma_k$  и  $\mu_k$  — управляющие параметры итерационной процедуры (2), (3),  $p(t, x)$  — решение системы (1) с начальным условием  $p(0, x) = x$ ; вектор  $a \in \mathbb{R}^N$ ; число  $b \in \mathbb{R}$ , через  $\langle \cdot, \cdot \rangle$  обозначено скалярное произведение в евклидовом пространстве  $\mathbb{R}^N$ . Было доказано (см. основную теорему работы [1]), что если управляющие параметры  $\gamma_k$  и  $\mu_k$  достаточно малы положительные числа, то последовательность  $\{x_k, t_k\}$  сходится по норме к искомому циклу и его периоду. Однако для практического применения предложенной процедуры важно знать, каким именно оценкам должны удовлетворять параметры  $\gamma_k$  и  $\mu_k$ . Цель настоящей работы состоит в получении этих оценок, исходя из свойств правой части системы

(1). Итерационный алгоритм (2), (3) был опубликован в тезисах [2–4].

### 1. ОСНОВНАЯ ТЕОРЕМА

Пусть  $\Gamma$  — изолированный цикл системы (1) с периодом  $T^*$ ,  $x^* \in \Gamma$  и

$$\langle a, x^* \rangle = b. \quad (4)$$

Рассмотрим гиперплоскость  $\pi$ , определяемую уравнением (4). Изменяя параметры  $a$  и  $b$ , всегда можно добиться трансверсальности плоскости  $\pi$  и цикла  $\Gamma$ .

Пусть  $R_0$  таково, что шар  $\|x\|_{\mathbb{R}^N} \leq R_0$  содержит отыскиваемый цикл,  $T_0 > T^*$  — некоторое число и, кроме того, для некоторых постоянных  $R_1, M_1, M_2, N_1$  и  $N_2$

$$\begin{aligned} \max\{\|p(t, x)\| : \|x\| \leq R_0, \quad 0 \leq t \leq T_0\} &\leq R_1, \\ \max_{\|x\| \leq R_1} \|f(x)\| &\leq M_1, \quad \max_{\|x\| \leq R_1} \|f'(x)\| &\leq M_2, \end{aligned} \quad (5)$$

$$\begin{aligned} \|f(x_1) - f(x_2)\| &\leq N_1 \|x_1 - x_2\|, \quad \|x\| \leq R_1, \\ \|f'(x_1) - f'(x_2)\| &\leq N_2 \|x_1 - x_2\|, \quad \|x\| \leq R_1. \end{aligned} \quad (6)$$

Тогда справедлива следующая

**Теорема.** В условиях основной теоремы работы [1] управляющие параметры процедуры (2), (3) могут быть выбраны из условий:  $0 < \alpha_0 \leq \gamma_k \leq \beta_0 \leq 1$ ,  $0 < \alpha_1 \leq \mu_k \leq \beta_1 \leq 1$ , где  $\beta_0$  и  $\beta_1$  — любые положительные числа, удовлетворяющие соотношениям

$$\frac{1}{2} K_1 \beta_0 + \frac{1}{4} (K_2 + L_1) \beta_1 < 1, \quad (7)$$

$$\frac{1}{2} L_2 \beta_1 + \frac{1}{4} (K_2 + L_1) \beta_0 < 1. \quad (8)$$



где

$$K_1 = 1 + \|a\|^2 + e^{N_1 T_0} + e^{M_2 T_0} + e^{(M_2 + N_1) T_0} + \frac{N_2(R_0 + R_1)(e^{(N_1 + M_2) T_0} - 1)e^{M_2 T_0}}{N_1 + M_2}, \quad (9)$$

$$K_2 = M_1 + ((R_0 + R_1)M_2 + M_1)e^{M_2 T_0}, \quad (10)$$

$$L_1 = N_1(R_0 + R_1)e^{N_1 T_0} + M_1(1 + e^{N_1 T_0}), \quad (11)$$

$$L_2 = M_1(R_0 N_1 + R_1 N_1 + M_1). \quad (12)$$

Доказательство теоремы см. в Приложении.

### 2. ЗАМЕЧАНИЯ

Напомним, что упомянутые в формулировке условия сводятся к следующему: точка  $x^*$  пересечения цикла  $\Gamma$  с плоскостью  $\pi$  и период  $T^*$  должны быть изолированными решениями системы уравнений

$$\begin{cases} (I - V^T(t, x))(x - p(t, x)) + (\langle a, x \rangle - b)a = 0 \\ \langle f(p(t, x)), p(t, x) \rangle = 0. \end{cases}$$

которая, в свою очередь, эквивалентна условию равенства нулю градиента функции погрешности

$$W(x, t) = \frac{1}{2} (\|x - p(t, x)\|^2 + (\langle a, x \rangle - b)^2). \quad (13)$$

Функция (13) неотрицательна. В ее нулях градиент обращается в нуль. Заметим: наряду с точкой  $\{x^*, T^*\}$  экстремалью (и нулем) функции (13) будет также точка  $\{p(T_1^*, x^*), T^*\}$ ,  $T_1^* < T^*$ , где  $p(T_1^*, x^*)$  — второе (из нескольких возможных) пересечение гиперплоскости  $\pi$  с циклом  $\Gamma$ . В силу трансверсальности и конечности периода  $T^*$ , все прочие пересечения плоскости  $\pi$  с циклом  $\Gamma$   $\varepsilon$ -отделены от точки  $x^*$ .

Поясним структуру функции погрешности. Первое слагаемое в правой части равенства (13) фактически означает невязку  $t$ -периодичности при фиксированном  $t$ . Второе слагаемое добавлено с целью выделить начальное условие для точки  $x^*$  на цикле  $\Gamma$ : без него многообразие минимумов функции  $W$  оказалась бы вся кривая  $\Gamma$ . Этот прием тесно связан с методом функционализации параметра при вычислении функциональной характеристики цикла (см. работы [5, 6]).

### 3. ПРИЛОЖЕНИЕ

**Доказательство теоремы.** Для доказательства нам понадобятся пять лемм.

**Лемма 1.** Пусть  $\|x_1\|, \|x_2\| \leq R_0$  и  $0 \leq t \leq T_0$ . Тогда

$$\|p(t, x_1) - p(t, x_2)\| \leq \|x_1 - x_2\| e^{N_1 t}. \quad (14)$$

**Доказательство.** Так как  $p(t, x_1) = x_1 + \int_0^t f(p(\tau, x_1)) d\tau$ ,  $p(t, x_2) = x_2 + \int_0^t f(p(\tau, x_2)) d\tau$ , то  $\|p(t, x_1) - p(t, x_2)\| \leq \|x_1 - x_2\| + N_1 \int_0^t \|p(\tau, x_1) - p(\tau, x_2)\| d\tau$ .

Из последнего неравенства и леммы Гронуолла—Беллмана вытекает оценка (14). ♦

**Лемма 2.** Пусть  $\|x\| \leq R_0$ ,  $0 \leq t_1, t_2 \leq T_0$ . Тогда

$$\|p(t_1, x) - p(t_2, x)\| \leq |t_1 - t_2| M_1. \quad (15)$$

**Доказательство.** Так как  $p(t_1, x) = x + \int_0^{t_1} f(p(\tau, x)) d\tau$ ,  $p(t_2, x) = x + \int_0^{t_2} f(p(\tau, x)) d\tau$ , то  $p(t_1, x) - p(t_2, x) = \int_{t_1}^{t_2} f(p(\tau, x)) d\tau$  и  $\|p(t_1, x) - p(t_2, x)\| \leq \max_{t_1 \leq \tau \leq t_2} \|f(p(\tau, x))\| \times |t_1 - t_2| \leq M_1 |t_1 - t_2|$ . ♦

**Лемма 3.** Пусть  $\|x\| \leq R_0$ ,  $0 \leq t \leq T_0$ . Тогда

$$\|V(t, x)\| \leq e^{M_2 t}. \quad (16)$$

**Доказательство.** Так как  $V(t, x) = I + \int_0^t f'_x(p(\tau, x)) V(\tau, x) d\tau$  то  $\|V(t, x)\| \leq 1 + M_2 \int_0^t \|V(\tau, x)\| d\tau$ .

Из последней оценки и леммы Гронуолла—Беллмана следует оценка (16). ♦

**Лемма 4.** Пусть  $\|x\| \leq R_0$ ,  $0 \leq t_1, t_2 \leq T_0$ . Тогда

$$\|V(t_1, x) - V(t_2, x)\| \leq M_2 e^{M_2 T_0} |t_1 - t_2|. \quad (17)$$

**Доказательство** аналогично доказательству леммы 2.

**Лемма 5.** Пусть  $\|x_1\|, \|x_2\| \leq R_0$ ,  $0 \leq t \leq T_0$ . Тогда

$$\begin{aligned} & \|V(t, x_1) - V(t, x_2)\| \leq \\ & \leq \frac{N_2}{N_1 + M_2} (e^{(N_1 + M_2) T_0} - 1) e^{M_2 t} \|x_1 - x_2\|. \end{aligned} \quad (18)$$

**Доказательство** аналогично доказательству леммы 1; используется лемма Гронуолла—Беллмана.

При доказательстве основной теоремы в работе [1] было показано, что сходимость  $\lim_{k \rightarrow \infty} (\|x_k - x^*\| + |t_k - T^*|) = 0$  имеет место, если только константы  $\beta_0$  и  $\beta_1$  подчинены неравенствам (7) и (8), где числа  $K_1, K_2, L_1$  и  $L_2$  опреде-

ляются из условий липшицевости градиента функции  $W(x, t)$ :

$$\|\nabla_t W(x_1, t_1) - \nabla_t W(x_2, t_2)\| \leq L_1 \|x_1 - x_2\| + L_2 |t_1 - t_2|,$$

$$\|\nabla_x W(x_1, t_1) - \nabla_x W(x_2, t_2)\| \leq K_1 \|x_1 - x_2\| + K_2 |t_1 - t_2|.$$

Цель ближайших выкладок — вычисление констант  $K_1, K_2, L_1$  и  $L_2$  исходя из значений  $R_0, R_1, M_1, M_2, N_1$  и  $N_2$ .

Легко видеть, что  $\nabla_x W(x, t) = (I - V^T(t, x))(x - p(t, x)) + \langle (a, x) - b, a \rangle$ ,  $\nabla_t W(x, t) = -\langle f(p(t, x)), x - p(t, x) \rangle$ .

Используя эти выражения и оценки из доказанных лемм (14)—(18) с помощью несложных преобразований получим неравенство:

$$\begin{aligned} \|\nabla_x W(x_1, t_1) - \nabla_x W(x_2, t_2)\| \leq & \left( 1 + \|a\|^2 + e^{N_1 T_0} + e^{M_2 T_0} + \right. \\ & \left. + \frac{N_2(R_0 + R_1)(e^{(N_1 + M_2)T_0} - 1)e^{M_2 T_0}}{N_1 + M_2} + e^{(M_2 + N_1)T_0} \right) \times \\ & \times \|x_1 - x_2\| + (M_1 + (R_0 M_2 + M_1 + R_1 M_2)e^{M_2 T_0}) |t_1 - t_2|. \end{aligned}$$

Таким образом, градиент  $\nabla_x W(x, t)$  по переменной  $x$  функции (13) удовлетворяет условию Липшица по  $x$  и  $t$  с константами  $K_1$  и  $K_2$ , причем последние имеют вид (9) и (10). Аналогично получаем

$$\begin{aligned} \|\nabla_t W(x_1, t_1) - \nabla_t W(x_2, t_2)\| \leq \\ \leq (R_0 N_1 e^{N_1 T_0} + M_1 + R_1 N_1 e^{N_1 T_0} + M_1 e^{N_1 T_0}) \|x_1 - x_2\| + \\ + (R_0 N_1 M_1 + R_1 N_1 M_1 + M_1^2) |t_1 - t_2|. \end{aligned}$$

Поэтому производная  $\nabla_t W(x, t)$  функции (13) удовлетворяет условию Липшица по  $x$  и  $t$  с константами  $L_1$  и  $L_2$ , определяемыми выражениями (11) и (12). Теорема доказана. ♦

### ЗАКЛЮЧЕНИЕ

Доказанная теорема носит прикладной характер и может оказаться полезной для непосредственных вычислений на ЭВМ. Задачи, связанные с поиском неустойчивых циклов, часто встречаются в нелинейной динамике. Например при исследовании уравнений Дуф-

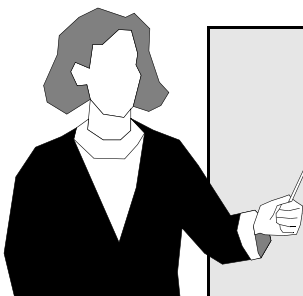
финга, Лоренца, Эно. Кроме того, в работах обширного класса для строгих аналитических доказательств применяется численный эксперимент. Суть подобных методов зачастую сводится к тому, чтобы установить отличие от нуля вращения поля, связанного с функцией последования (отображением Пуанкаре). Иногда это удается сделать благодаря устойчивости вращения к малым возмущениям поля, а следовательно, и к вычислительным ошибкам. В свете этого приведенные выше алгоритм и оценки параметров можно использовать для первоначальной локализации цикла. Впоследствии может оказаться, что на некоторой сфере с центром в точке  $x^*$  на секущей гиперплоскости вращение поля функции последования не равно нулю и, таким образом, станет возможным строго доказать существование цикла в окрестности найденного приближения.

### ЛИТЕРАТУРА

1. *Исмаилов И. Г.* Об одном итерационном алгоритме построения циклов автономных систем // Проблемы управления. — 2005. — № 3. — С. 10—12. ([www.ipu.ru/period/pu](http://www.ipu.ru/period/pu)).
2. *Бобылев Н. А., Исмаилов И. Г., Коровин С. К.* Об одном алгоритме построения предельных циклов в системах автоматического регулирования // IV междунар. семинар “Устойчивость и колебания нелинейных систем управления”. — М. — 1996. — С. 6.
3. *Ismailov I. G.* On the scheme of approximate construction of cycles of nonlinear systems // Fourth International conference on “Control, automation, robotics and vision”. — Singapore, — 1996.
4. *Ismailov I. G.* On the approximate construction of cycles in automatic control systems // Fourth International Symposium on “Method and Models in Automation and Robotics”. — Poland, — 1997.
5. *Бобылев Н. А., Красносельский М. А.* Функционализация параметра и теорема родственности для автономных систем // Дифференциальные уравнения. — 1970. — № 11. — С. 1946—1952.
6. *Бобылев Н. А., Коровин С. К.* Теоремы родственности в теории нелинейных колебаний // Методы анализа нелинейных систем: Сб. науч. тр. — МГУ, Ин-т системного анализа РАН. — М., — 1997.

☎ (095) 334-79-00

E-mail: [ilham\\_is@mail.ru](mailto:ilham_is@mail.ru)



### Вниманию подписчиков!

В каталоге “Роспечати” на 2005 г. ошибочно указана периодичность журнала “Проблемы управления” — 4 номера в год. Однако с 2005 г. мы выходим **6 раз в год**. Если Вы подписались по каталогу “Роспечати”, то для получения № 3 и 6 Вам необходимо на них подписаться по объединенному каталогу “Пресса России” (индекс **38006**) или через Редакцию.

Редакция



# СПОСОБ САМОДИАГНОСТИРОВАНИЯ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ

В. А. Ведешенков

*Институт проблем управления им. В. А. Трапезникова, г. Москва*

Предложен способ самодиагностирования (СД) компонентов неоднородных цифровых систем, содержащих одинаковое число устройств двух типов (процессоров, памяти). Для осуществления контроля и СД связанные разнотипные устройства объединяются в подсистемы: проверяющие и проверяемые. Допускаются устойчивые кратные отказы ограниченного числа компонентов. Разработана организация СД рассматриваемых неоднородных цифровых систем, основанная на постепенном расширении множества исправных модулей. Дан пример реализации СД компонентов в 14-модульной системе.

## ВВЕДЕНИЕ

Любой метод системного диагностирования технического состояния компонентов в цифровых системах (многопроцессорных и многомашинных вычислительных системах (ММВС)) основан на том, что вводится ограничение на максимальное число произвольно расположенных отказавших компонентов, которое позволяет оставшимся исправным модулям получить информацию, достаточную для правильного диагностирования состояния компонентов. Для описания результатов тестирования модулей в данной работе используется модель Барси—Грандони—Маестрини (БГМ) [1], которая предполагает, что модули, на которые разбивается исходная ММВС, однородные и обладают функциональными возможностями, достаточными для реализации диагностических алгоритмов, составными частями которых являются процедуры проверки компонентов системы и сбора результатов этих проверок.

Однако многие ММВС содержат устройства различных типов (процессоры, станции памяти, каналы связи). Примером такой ММВС может служить неоднородная система обработки радиолокационной информации [2], содержащая семь процессоров и семь станций памяти, соединенных каналами связи в соответствии с графом, показанным на рис. 1, где процессоры обозначены четными номерами, а станции памяти — нечетными. Любой процессор системы связан с любым другим процессором через одну станцию памяти. Наличие устройств разных типов не позволяет непосредственно применить системные методы диагностирования. Но автоматизированное диагностирование состояния компонентов не-

однородных цифровых систем представляется необходимым, так как позволяет сократить время восстановления работоспособности систем после обнаружения факта их неправильного функционирования.

Далее представлен способ организации контроля и самодиагностирования (СД) состояния компонентов неоднородных цифровых систем, предполагающий на

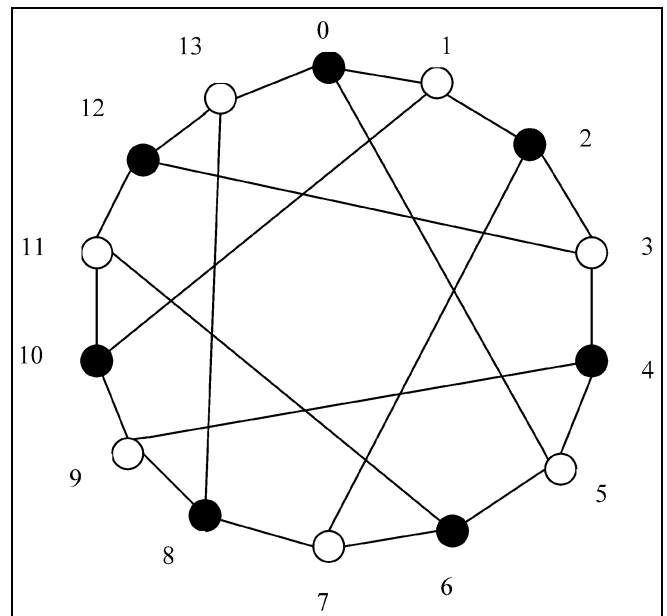


Рис. 1. Граф неоднородной системы  $Q_{14, 21}$

этапах контроля и СД объединение связанных разнотипных устройств в проверяющие и проверяемые подсистемы, обладающие необходимыми функциональными возможностями для реализации диагностических алгоритмов.

Напомним, что в однородных системах при выполнении процессов контроля и СД проверяющий модуль подает на входы проверяемого модуля последовательность проверок (тест), получает ответные реакции проверяемого модуля на этот тест, сравнивает их с эталонными, а результат сравнения оценивает в двоичной форме: 0 — результаты правильные, 1 — неправильные. В модели БГМ предполагается, что оценка результатов теста всегда достоверна, если исправен проверяющий модуль; эта оценка равна 1, если неисправны оба модуля, и недостоверна (0 или 1) в том случае, когда неисправен проверяющий модуль, а проверяемый исправен. Поэтому нулевая оценка, выдаваемая проверяющим модулем, с большой вероятностью подтверждает исправность проверенного им модуля. Такая предпосылка модели БГМ существенно упрощает процедуру диагностирования, так как для идентификации исправных модулей достаточно 0-путей единичной длины (в терминологии работы [3]).

Суммарная оценка результатов выполненных тестов  $t_1, \dots, t_l$  образует двоичный вектор  $R = r_1, \dots, r_l$ , который называется реальным синдромом. В большинстве алгоритмов системного диагностирования предполагается (или констатируется), что результаты тестов, образующие реальный синдром  $R$ , собираются и обрабатываются (дешифрируются) внешним устройством. По результатам работы этого устройства (результатам диагностирования) обслуживающий персонал заменяет отказавшие компоненты и восстанавливает работоспособность ММВС. В настоящей статье, как и в работах [4, 5], развивается другой подход, основанный на организации самодиагностирования системы, когда необходимые алгоритмы, включая сбор реального синдрома  $R$  и его обработку, реализуются на исправных процессорах самой системы.

## 1. ПОСТАНОВКА ЗАДАЧИ

Пусть анализируемая система представлена неоднородным диагностическим графом  $Q_{N, L}$ ,  $N = 2n$  вершин которого отображают  $n$  активных устройств (процессоров) и  $n$  пассивных устройств (памятей) системы, а  $L$  ненаправленных дуг между вершинами — линии связи между активными и пассивными устройствами (процессорами и памятьями). Будем считать, что каждая вершина одного типа связана ненаправленными дугами с  $p$  вершинами другого типа, тогда  $L = pn$ .

Будем считать, что допускаются устойчивые отказы ограниченного числа вершин и дуг, причем их неисправности таковы, что прекращают работу модулей, в которых они возникают, и не влияют на работоспособность смежных модулей. Предполагается также, что во время выполнения процесса СД новые отказы не возникают. Для описания результатов тестирования компонентов применяется модель БГМ. Предполагается, что в исходной системе есть диагностический монитор, который организует процессы контроля и СД.

Требуется разработать такой способ организации контроля и СД компонентов неоднородной системы, чтобы после окончания СД каждое исправное активное устройство (процессор) имело одинаковую информацию для идентификации состояния остальных компонентов (процессоров, памяти и линий связи) системы.

## 2. СПОСОБ САМОДИАГНОСТИРОВАНИЯ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ

Процесс СД компонентов однородных систем включает в себя [4, 5]:

— исполнение исправным модулем функций проверяющего;

— нахождение пути от одного исправного модуля к другому исправному для передачи ему функций проверяющего модуля;

— обработку результатов выполненных проверок.

Для реализации подобных процедур в неоднородных системах приведенного типа построим проверяющие и проверяемые подсистемы следующим образом. В проверяющую подсистему  $U_{i,j}$  включим устройство пассивного типа (память  $u_i$ ), устройство активного типа (процессор  $m_j$ ) и линию связи  $l_{i-j}$  между ними, а в проверяемую подсистему  $U_{j,h,k}$  включим память  $u_h$ , процессор  $m_k$ , линию связи  $l_{h-k}$  между ними, а также линию связи  $l_{j-h}$ , соединяющую процессор  $m_j$  (из подсистемы  $U_{i,j}$ ) и память  $u_h$ . (Отметим, что номера  $(i, j, h, k)$ , записанные в индексах подсистем  $U_{i,j}$ ,  $U_{j,h,k}$ , определяют номера  $i, h$  памяти и  $j, k$  процессоров, входящих в эти подсистемы.) При объединении различных устройств неоднородной системы  $Q_{14, 21}$  (см. рис. 1) в подсистемы приведенного состава используем следующий прием: каждую память  $u_i$  ( $i = 0, 1, \dots, n - 1$ ) объединим с  $p$  процессорами  $m_{j_1}, \dots, m_{j_p}$ , непосредственно связанными с этой памятью  $u_i$ , и образуем проверяющие подсистемы  $U_{i,j_1}, \dots, U_{i,j_p}$ . В результате таких объединений для рассматриваемой системы получим  $np$  (т. е. 21) подсистем  $U_{i,j}$ , различающихся номерами входящих в них устройств памяти и/или процессоров. В табл. 1 для системы  $Q_{14, 21}$  приведен состав семи (из 21-й возможных) проверяющих подсистем  $U_{i,j}$  ( $M_j$ ) и 14-ти (из 42-х возможных) проверяемых ими подсистем  $U_{j,h_1,k_1}$  ( $M_{j,h_1,k_1}$ ),  $U_{j,h_2,k_2}$  ( $M_{j,h_2,k_2}$ ); (в скобках показаны более компактные символы подсистем, используемые в примере).

Проверка подсистемы  $U_{j,h,k}$  (с помощью теста  $t_{j,h,k}$ ) выполняется следующим образом. Получив управление (в процессе контроля или СД), процессор  $m_j$  подсистемы  $U_{i,j}$  передает по линии  $l_{j-h}$  в память  $u_h$  команду на запуск процесса самоконтроля в подсистеме  $U_{j,h,k}$ . Процессор  $m_k$ , выбрав из памяти  $u_h$  и расшифровав эту команду, проводит самоконтроль подсистемы  $U_{j,h,k}$  с помощью тестовых программ, хранящихся в памяти этой подсистемы. (Естественно, что при этом не проверяется линия  $l_{j-h}$ , соединяющая подсистемы  $U_{i,j}$  и  $U_{j,h,k}$ , но ее состояние влияет на информацию, передаваемую процессором  $m_j$  в память  $u_h$  и принимаемую из нее). Резуль-



Таблица  $T_S$  состава проверяющих и проверяемых подсистем

Строка $T_{i,j}(T_j)$	Проверяющая подсистема	Проверяемые подсистемы и оценки их проверок			
	$U_{i,j}(M_j)$	$U_{j,h1,k1}(M_{j,k1})$	$r_{j,h1,k1}(r_{j,k1})$	$U_{j,h2,k2}(M_{j,k2})$	$r_{j,h2,k2}(r_{j,k2})$
$T_{1,2}(T_0)$	1, (1–2), 2 ( $M_0$ )	(2–3), 3, (3–4), 4 ( $M_{0,1}$ )		(2–7), 7, (7–8), 8 ( $M_{0,3}$ )	
$T_{3,4}(T_1)$	3, (3–4), 4 ( $M_1$ )	(4–5), 5, (5–6), 6 ( $M_{1,2}$ )		(4–9), 9, (9–10), 10 ( $M_{1,4}$ )	
$T_{5,6}(T_2)$	5, (5–6), 6 ( $M_2$ )	(6–7), 7, (7–8), 8 ( $M_{2,3}$ )		(6–11), 11, (11–12), 12 ( $M_{2,5}$ )	
$T_{7,8}(T_3)$	7, (7–8), 8 ( $M_3$ )	(8–9), 9, (9–10), 10 ( $M_{3,4}$ )		(8–13), 13, (13–0), 0 ( $M_{3,6}$ )	
$T_{9,10}(T_4)$	9, (9–10), 10 ( $M_4$ )	(10–11), 11, (11–12), 12 ( $M_{4,5}$ )		(10–1), 1, (1–2), 2 ( $M_{4,0}$ )	
$T_{11,12}(T_5)$	11, (11–12), 12 ( $M_5$ )	(12–13), 13, (13–0), 0 ( $M_{5,6}$ )		(12–3), 3, (3–4), 4 ( $M_{5,1}$ )	
$T_{13,0}(T_6)$	13, (13–0), 0 ( $M_6$ )	(0–1), 1, (1–2), 2 ( $M_{6,0}$ )		(0–5), 5, (5–6), 6 ( $M_{6,2}$ )	

таты выполненного теста  $t_{j,h,k}$  процессор  $m_k$  записывает в ячейки памяти  $u_h$ , откуда их, по истечению определенного времени, считывает процессор  $m_j$ , сравнивает с эталонами из собственной памяти  $u_i$  и формирует оценку  $r_{j,h,k}$ : 0, если результаты совпали с эталонами, 1 — при несовпадении, в том числе, из-за искажений, внесенных линией  $l_{j-h}$ . В дальнейшем эти действия кратко будем называть так: подсистема  $U_{i,j}$  проверяет подсистему  $U_{j,h,k}$  и формирует оценку  $r_{j,h,k}$  результатов тестирования. Выполнив тесты для всех сформированных подсистем, получим реальный синдром  $R$ , объединяющий оценки проведенных тестов.

Таким образом, предлагаемый способ СД неоднородных систем основан на объединении разнотипных устройств исходной системы в подсистемы с функциональными возможностями, достаточными для выполнения процедур самоконтроля и других диагностических процедур. Далее описана реализация этого способа, основанная на методе СД для однородных систем из работ [4, 5].

**3. ОРГАНИЗАЦИЯ САМОДИАГНОСТИРОВАНИЯ КОМПОНЕНТОВ НЕОДНОРОДНЫХ ЦИФРОВЫХ СИСТЕМ**

Контроль работоспособности однородной цифровой системы заключается в проверке каждым проверяющим модулем технического состояния своих приемников и соответствующих линий связи. Простейший способ выполнения проверок одного модуля разными проверяющими модулями — последовательный. Для его оптимальной реализации на диагностическом графе однородной системы нужно построить гамильтонов цикл, который обеспечивает однократный обход всех модулей исправной системы. В неоднородных системах функции проверяющих модулей исполняют проверяющие подсистемы  $U_{i,j}$ , включающие в себя активные устройства  $m_j$  (процессоры в системе  $Q_{14,21}$ ). Поэтому в неоднородных системах нужно строить цикл, проходящий через все проверяющие подсистемы.

Исходную информацию о составе выбранных проверяющих и проверяемых подсистем представим в виде

таблицы состава  $T_S$ , в каждой строке  $T_{i,j}$  которой записан состав проверяющей подсистемы  $U_{i,j}$  проверяемых ею подсистем  $U_{j,h1,k1}, U_{j,h2,k2}, \dots$ ; также есть поля для записи оценок проверок этих подсистем  $r_{j,h1,k1}, r_{j,h2,k2}, \dots$

Пример такой таблицы, построенной для системы  $Q_{14,21}$  (см. рис. 1), показан в табл. 1. Номера  $i, j, h, k$ , записанные в клетках этой таблицы, определяют номера памяти  $u_i, u_h$  и процессоров  $m_j, m_k$ , участвующих в реализации теста  $t_{j,h,k}$ , а пары чисел в скобках определяют номера линий, связывающих эти памяти и процессоры. Подчеркнем, что структура графа  $Q_{14,21}$  позволяет для каждой проверяющей подсистемы построить четыре различных проверяемых подсистемы. Однако в каждой строке табл. 1 показаны только две подсистемы, в составе которых и проверяются линии связи, смежные с процессором каждой проверяющей подсистемы.

Для отображения местонахождения проверяющих подсистем в цикле контроля будем записывать в строке  $T_{i,j}$  подсистемы в таком порядке, что именно проверяющая подсистема  $U_{h1,k1}$  (“ядро” проверяемой подсистемы  $U_{j,h1,k1}$ ) является в построенном цикле соседней справа для проверяющей подсистемы  $U_{i,j}$ . Например, в строке  $T_{3,4}$  табл. 1 подсистема  $U_{3,4}$  проверяет подсистемы  $U_{4,5,6}$  и  $U_{4,9,10}$ , и в следующей строке  $T_{5,6}$  в качестве проверяющей записана подсистема  $U_{5,6}$  (“ядро” проверяемой подсистемы  $U_{4,5,6}$ ), являющаяся в цикле контроля соседом справа для подсистемы  $U_{3,4}$ .

Традиционно процесс контроля запускается диагностическим монитором (ДМ) через сравнительно небольшие интервалы времени, что препятствует накоплению отказавших компонентов в системе.

Назначим в качестве начальной подсистемы процесса контроля одну из подсистем, обозначим ее  $U_{i,n}$ . Для минимизации числа повторных пусков системы контроля с участием человека-оператора целесообразно назначить в качестве начальной ту подсистему, в которой произведена замена отказавшего компонента, обнаруженного в последнем цикле СД. Этот компонент имеет

наименьшее время работы в системе и, следовательно, наименьшую вероятность отказа.

По сигналу ДМ о начале процесса контроля подсистема  $U_{i,n}$  выполняет следующие действия:

- из таблицы  $T_S$  выделяет строку  $T_{i,n}$ , находит в ней индексы  $(n, h1, k1)$ ,  $(n, h2, k2)$ , ... проверяемых подсистем  $U_{n,h1,k1}$ ,  $U_{n,h2,k2}$ , ...; проверяет эти подсистемы и формирует оценки  $r_{n,h1,k1}$ ,  $r_{n,h2,k2}$ , ..., равные 0, если результаты тестов совпали с эталонами, и равные 1 — при несовпадении, в том числе, из-за искажений, внесенных линиями связи  $U_{i,n}$  с подсистемами  $U_{n,h1,k1}$ ,  $U_{n,h2,k2}$ , ...;
- сформированные оценки записывает в соответствующие клетки строки  $T_{i,n}$ , анализирует их:
  - если все оценки — нулевые ( $r_{n,h1,k1} = 0$ ,  $r_{n,h2,k2} = 0$ , ...), то через память  $u_{h1}$  передает подсистеме  $U_{h1,k1}$  (“ядру” подсистемы  $U_{n,h1,k1}$ ) управление продолжением контроля;
  - если среди полученных оценок есть как единичные, так и нулевые (например,  $r_{n,h2,k2} = 0$ ), то передает одной из подсистем с нулевой оценкой (например, подсистеме  $U_{h2,k2}$ ) управление для продолжения процесса контроля;
  - если тесты самоконтроля всех проверявшихся подсистем дали единичные оценки, то ДМ запускает процесс контроля с другой проверяющей подсистемы (например, с подсистемы  $U_{i,n2}$ , находящейся в “середине” выбранного цикла контроля); ДМ запускает процесс контроля с подсистемы  $U_{i,n2}$  и в том случае, если из-за неисправности начальной подсистемы  $U_{i,n}$  процесс контроля остановился и к заданному сроку ДМ не получил сигнала об окончании СД.

Подсистема  $U_{h1,k1}$  ( $U_{h2,k2}$ ) и каждая из последующих подсистем — “соседей” по циклу контроля — выполняют аналогичные действия в случае получения нулевых оценок. Этот процесс проверок и “хождения” по вершинам выбранного цикла заканчивается, когда управление возвращается к начальной подсистеме  $U_{i,n}$  ( $U_{i,n2}$ ). Если же на одном из этапов контроля (кроме начального) получена хотя бы одна единичная оценка, свидетельствующая об обнаружении одного или нескольких отказавших компонентов, то процесс контроля прерывается, и система переходит к процессу СД отказавших компонентов, который включает в себя:

- исполнение исправной подсистемой функций проверяющей;
- нахождение пути передачи управления и информации от одной исправной проверяющей подсистемы к другой;
- обработку результатов выполненных тестов.

На основе информации, полученной в процессе контроля, ДМ запускает процесс СД с той подсистемы  $U_{i,j}$ , которая первой получила хотя бы одну единичную оценку при проверке состояний связанных с ней подсистем  $U_{j,h1,k1}$ ,  $U_{j,h2,k2}$ , ...; естественно, подсистема  $U_{i,j}$  должна быть исправной и проверенной, в частности,

она не может быть начальной подсистемой процесса контроля  $U_{i,n}$  ( $U_{i,n2}$ ).

Начальная подсистема  $U_{i,j}$  ( $j \neq n, n2$ ) выполняет следующие действия:

- формирует промежуточную таблицу  $PT_{i,j}$ , переписывая в нее строку  $T_{i,j}$  с оценками, полученными в процессе контроля (в индексе символа данной таблицы указан индекс проверяющей подсистемы  $U_{i,j}$ );
- по нулевым оценкам в строке  $T_{i,j}$  выделяет подмножество индексов исправных подсистем  $\{q_{i,j}\}$  и одной из этих подсистем (первой в строке  $T_{i,j}$ , если она исправна) передает таблицу  $PT_{i,j}$  и управление процессом СД.

Если среди подсистем  $U_{j,h1,k1}$ ,  $U_{j,h2,k2}$ , ..., проверенных подсистемой  $U_{i,j}$ , нет ни одной исправной, то процесс СД не может быть продолжен, и для получения диагностических результатов ДМ должен повторить запуск процесса контроля с другой начальной подсистемы (например, с подсистемы  $U_{i,n2}$ ).

Очередная исправная подсистема  $U_{i,v}$ , получив управление процессом СД, выполняет следующие действия:

- 1) из таблицы состава  $T_S$  выделяет строку  $T_{i,v}$ , переписывает ее в таблицу  $PT_{i,j}$ , полученную от предшественника — подсистемы  $U_{i,j}$  ( $U_{i,v-1}$ );
- 2) в строке  $T_{i,v}$  выделяет индексы проверяемых подсистем  $U_{v,h1,k1}$ ,  $U_{v,h2,k2}$ , ..., проверяет эти подсистемы и в клетки  $r_{v,h1,k1}$ ,  $r_{v,h2,k2}$ , ... строки  $T_{i,v}$  записывает оценки результатов выполненных тестов;
- 3) по нулевым оценкам проверок в строке  $T_{i,v}$  выделяет подмножество  $\{q_{i,v}\}$  индексов исправных “преемников” — проверенных подсистем, сравнивает индексы из подмножества  $\{q_{i,v}\}$  с индексами строк в таблице  $PT_{i,j}$ :
  - если один индекс (или больше) из подмножества  $\{q_{i,v}\}$  отсутствует в списке индексов строк в таблице  $PT_{i,j}$ , то передает подсистеме с этим индексом таблицу  $PT_{i,j}$  и управление СД;
  - если список  $\{q_{i,v}\}$  пуст или все индексы  $\{q_{i,v}\}$  — в списке индексов строк в таблице  $PT_{i,j}$ , то переход к п. 4;
- 4) находит одну из исправных подсистем  $U_{i,v+1}$ ,  $U_{i,v+2}$ , ...,  $U_{i,v+w}$ , еще не исполнявших функции проверяющей подсистемы, и передает ей таблицу  $PT_{i,j}$  и управление СД; если при выполнении этой процедуры выяснится, что все обнаруженные исправные подсистемы уже исполняли функции проверяющих, то переход к п. 5;
- 5) передает информацию, собранную в таблице  $PT_{i,j}$  подсистемы  $U_{i,v+w}$ , в направлении, обратном тому, по которому эта информация собиралась; получив идентичную информацию в виде таблицы  $PT_{i,j}$  от подсистемы  $U_{i,v+w}$ , каждая из исправных подсистем выполняет ее обработку и формирует одинаковый диагноз.

Для нахождения одной из исправных подсистем  $U_{i,v+1}$ ,  $U_{i,v+2}$ , ...,  $U_{i,v+w}$ , еще не исполнявших функций





проверяющей подсистемы, подсистема  $U_{i,v}$  находит индексы  $\{q_{i,v^*}\}$  исправных подсистем строки  $T_{i,v^*}$  и сравнивает их с индексами строк в таблице  $PT_{i,j}$ . Если при сравнении списков индексов окажется, что:

- один индекс (или больше) из подмножества  $\{q_{i,v^*}\}$  отсутствует в списке индексов строк в таблице  $PT_{i,j}$ , то подсистема  $U_{i,v}$  передает подсистеме с этим индексом  $(i, v + 1)$  таблицу  $PT_{i,j}$  и управление СД;
- список  $\{q_{i,v^*}\}$  пуст или все индексы  $\{q_{i,v^*}\}$  — в списке индексов строк в таблице  $PT_{i,j}$ , то переход к анализу предыдущей строки  $T_{i,v^*-1}$ .

Эти действия проводятся поочередно снизу вверх, начиная с нижней строки  $T_{i,v}$ . При нахождении исправной подсистемы  $U_{i,v+1}$  в строке  $T_{i,v^*}$  управление СД и таблица  $PT_{i,j}$  передаются этой подсистеме снизу вверх через проверяющие подсистемы  $U_{i,v^*}, \dots, U_{i,v}$ . Поиск прекращается, если при достижении верхней строки  $T_{i,j}$  окажется, что и в этой строке нет исправной подсистемы, не исполнявшей функций проверяющей подсистемы.

Для обработки информации, собранной в таблице  $PT_{i,j}$  подсистемы  $U_{i,v+w}$ , каждая исправная подсистема переписывает ее в диагностическую таблицу  $T_D$  следующего вида. В клетках левого столбца таблицы  $T_D$  перечислены компоненты проверяемых подсистем  $U_{j,h,k} (M_{j,k})$ , взятые из таблицы состава  $T_S$ , а клетки следующего столбца предназначены для записи оценок  $r_{j,h,k} (r_{j,k})$  проверки этих подсистем, полученных в процессе СД. Последующие столбцы таблицы  $T_D$  обозначены номерами компонентов системы, с точностью до которых диагностируются подсистемы  $M_k$  и линии  $L_{j-k}$  между подсистемами  $M_j$  и  $M_k$ . Вхождение компонентов  $M_k, L_{j-k}$  в состав подсистемы  $M_j$ , приведенной в левом столбце таблицы  $T_D$ , отмечается знаком \*, поставленным в клетке на пересечении соответствующего столбца и строки  $M_{j,k}$ . В нижнюю часть таблицы  $T_D$  добавлены три строки, озаглавленные (в левом столбце) символами  $A_0, A_1, A_z$  и предназначенные для записи результатов обработки полученной информации. Пример такой таблицы, построенной для системы  $Q_{14,21}$ , показан в табл. 2, в

Таблица 2

Диагностическая таблица  $T_D$  для Примера

Проверяемые подсистемы $M_{j,k}$	$r_{j,k}$	Компоненты системы																					
		$M_0$	$L0-1$	$M_1$	$L1-2$	$M_2$	$L2-3$	$M_3$	$L3-4$	$M_4$	$L4-5$	$M_5$	$L5-6$	$M_6$	$L6-0$	$L0-3$	$L1-4$	$L2-5$	$L3-6$	$L4-0$	$L5-1$	$L6-2$	
$M_{0,1}$	0		0	0																			
$M_{0,3}$	0							0								0							
$M_{1,2}$	0				0	0																	
$M_{1,4}$	1									1								1					
$M_{2,3}$	0						0	0															
$M_{2,5}$	0										0							0					
$M_{3,4}$	0								0	0													
$M_{3,6}$	1													1					1				
$M_{4,5}$	0									0	0												
$M_{4,0}$	0	0																		0			
$M_{5,6}$	1											1	1										
$M_{5,1}$	0			0																	0		
$M_{6,0}$	—	*													*								
$M_{6,2}$	—					*																	*
$A_0$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$A_1$														1			1						
$A_z$												Z		Z				Z				Z	

клетках второго столбца которой записаны оценки проверок для приводимого ниже Примера. Правила обработки записей таблицы  $T_D$  и интерпретации полученных результатов даны ниже.

Процедура обработки информации из таблицы  $PT_{i,j}$ , выполняемая каждой исправной подсистемой, состоит из следующих шагов:

- из строк  $T_{i,j}, \dots, T_{i,v+w}$  таблицы  $PT_{i,j}$  выделяются оценки проверок  $r_{j,h1,k1}, r_{j,h2,k2}, \dots, r_{v+w,h1,k1}$ , и записываются в соответствующие клетки столбца  $r_{j,h,k}$  таблицы  $T_D$ , оставшиеся клетки этого столбца заполняются прочерками (—);
- “размножаются” 0 и 1, записанные в столбце  $r_{j,h,k}$  путем записи идентичных значений (0 или 1) в клетки той же строки  $T_{i,j}$ , отмеченные знаком \*;
- заполняются прочерками (—) клетки строк  $A_0, A_1, A_Z$ ;
- выполняется следующая обработка записей таблицы  $T_D$ :
  - в клетку на пересечении строки  $A_0$  и  $k$ -го столбца записывается 0, если в вышележащих клетках этого столбца есть хотя бы один нуль;
  - в клетку на пересечении строки  $A_1$  и столбца, озаглавленного символом одной из вершин графа (памяти  $u_h$ , процессора  $m_k$ , подсистемы  $M_k$ ), записывается 1, если в вышележащих клетках этого столбца есть хотя бы одна единица и нет ни одного нуля;
  - обрабатываются записи оставшихся столбцов, озаглавленных символами линий связи  $l_{j-h}, l_{h-k}$  ( $l_{j-k}$ ) и содержащих только единицы, следующим образом:
    - если обе вершины, которые связывает рассматриваемая линия, исправны (в соответствующих клетках строки  $A_0$  записаны 0), то в клетку на пересечении строки  $A_1$  и данного столбца записывается 1;
    - если хотя бы одна из вершин, которые связывает рассматриваемая линия, подозревается в отказе (в соответствующей клетке строки  $A_1$  записана 1), то в клетку на пересечении строки  $A_Z$  и данного столбца записывается Z;
- расшифровываются записи строк  $A_0, A_1$  и  $A_Z$ :
  - в соответствии с местоположением нулевых записей в строке  $A_0$  составляется список исправных компонентов (активных и пассивных устройств, линий связи);
  - в соответствии с местоположением единиц в строке  $A_1$  составляется список вершин (активных и пассивных устройств) и дуг (линий связи), подозреваемых в отказе;
  - в соответствии с местоположением символов Z в строке  $A_Z$  составляется список дуг (линий связи), состоянием которых маскируется неисправным состоянием хотя бы одной из вершин, которые она соединяет;
  - прочерки (—), оставшиеся в клетках на пересечении строк  $A_0, A_1$  и  $A_Z$  с  $k$ -м столбцом, означают, что полученной информации недостаточно для оп-

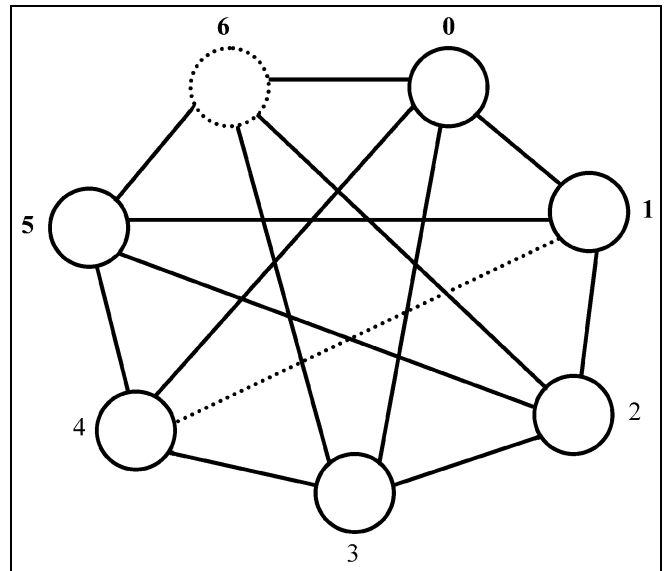


Рис. 2. Граф однородной системы  $Q_{7,14}$

ределения технического состояния данного компонента (вершины, дуги графа). ♦

Предлагаемая организация СД обеспечивает получение диагностических заключений при выполнении следующих требований к составу исправных компонентов:

- в системе должно быть не менее двух исправных проверяющих подсистем и исправная линия связи между ними;
- у каждой исправной подсистемы должна быть исправна хотя бы одна из  $p$  ее входных линий.

Так как восстановление работоспособности компонентов, подозреваемых в отказе, осуществляется после выполнения СД, то данную организацию СД следует отнести к методам однократного диагностирования. Поэтому для отказовых ситуаций, которым соответствуют неразличимые потенциальные синдромы, можно получить неполный, а иногда и неправильный диагноз.

#### ПРИМЕР

Особенности разработанной организации СД покажем на примере неоднородной системы [2] с графом  $Q_{14,21}$  (см. рис. 1). В работе [2] автор использовал неоднородную систему (см. рис. 1) для иллюстрации возможностей разработанного им алгоритма построения отказоустойчивых систем с реконfigurацией. Для этого он объединил две разнотипные вершины (память и процессор) исходного графа в одну вершину, названную агрегированной; в результате такого объединения граф неоднородной системы превратился в граф однородной системы  $Q_{7,14}$  (рис. 2). Обозначим объединенную (агрегированную) подсистему символом  $M_j$ , если она выполняет функции проверяющей подсистемы, и символом  $M_k$  при вхождении ее в проверяемую подсистему  $M_j, k$ , а линию связи между подсистемами  $M_j$  и  $M_k$  символом  $l_{j-k}$  ( $j \neq k = 0, \dots, 6$ ). В работе [2] выбран вариант, при



котором в подсистему  $M_j$  ( $M_k$ ) объединены смежные процессор и память, лежащие на выбранном гамильтоновом пути. Состав полученных модулей показан в столбце  $M_j$  табл. 1. При построении отказоустойчивых систем единицей резервирования считается агрегированная подсистема, поэтому естественно считать, что разрешающая способность диагностирования отказавших компонентов должна быть равна также одному заменяемому компоненту (подсистеме или линии связи) [2, 5]. Как отмечалось, в табл. 1 приведена таблица состава  $T_S$ , предназначенная для организации тестирования в системе  $Q_7, 14$ .

Для проведения контроля выберем цикл, охватывающий семь подсистем  $M_0, \dots, M_6$  в порядке возрастания их номеров, и при исправном состоянии всех компонентов управление процессом контроля от подсистемы  $M_6$  вернется к подсистеме  $M_0$ .

Предположим, что за время между сеансами контроля в системе накопились отказы подсистемы  $M_6$  и линии связи (1–4) (эти компоненты показаны пунктиром на рис. 2). По сигналу ДМ о начале контроля подсистема  $M_0$  выполнит следующие действия: из таблицы  $T_S$  (см. табл. 1) выделит строку  $T_0$ , найдет в ней индексы (0, 1) и (0, 3) проверяемых подсистем  $M_{0,1}$  и  $M_{0,3}$ , проверит их и сформирует оценки  $r_{0,1} = r_{0,3} = 0$ , которые запишет в соответствующие клетки строки  $T_0$  (табл. 3); так как обе оценки нулевые, то передаст подсистеме  $M_1$  (“ядру” проверенной исправной подсистемы  $M_{0,1}$ ) управление продолжением контроля (в строке  $T_0$  полужирным шрифтом выделен 0, по которому передается управление; таким обозначением будем пользоваться и в дальнейшем).

Подсистема  $M_1$ , действуя аналогично, проверит подсистемы  $M_{1,2}$  и  $M_{1,4}$ , получит нулевую и единичную оценки (показаны в строке  $T_1$  табл. 3) и перейдет к процессу СД (для сокращения числа таблиц будем записывать оценки, находящиеся в разных подсистемах, в общей табл. 3).

В начале процесса СД подсистема  $M_1$  выполнит следующие действия:

- сформирует промежуточную таблицу  $PT_1$  (в табл. 3), в которую переписет строку  $T_1$  с оценками, полученными в процессе контроля;
- по нулевой оценке ( $r_{1,2} = 0$ ) выделит номер 2 исправной подсистемы  $M_2$ , которой и передаст таблицу  $PT_1$  и управление продолжением процесса СД.

Подсистема  $M_2$ , выполнит следующие действия:

- из таблицы состава  $T_S$  (см. табл. 1) выделит строку  $T_2$ , переписет ее в таблицу  $PT_1$ ;
- в строке  $T_2$  выделит индексы (2, 3) и (2, 5) проверяемых подсистем  $M_{2,3}$  и  $M_{2,5}$ , проверит их и в соответствующие клетки строки  $T_2$  запишет оценки  $r_{2,3} = r_{2,5} = 0$ ;
- по нулевым оценкам выделит подмножество  $\{q_2 = 3, 5\}$  номеров исправных “преемников”, сравнит эти номера с индексами строк 1, 2 в таблице

$PT_1$ : так как эти числа не совпадают, то передаст подсистеме  $M_3$  таблицу  $PT_1$  и управление СД.

Подсистема  $M_3$ , а затем и каждая из подсистем  $M_4$  и  $M_5$  проверит своих преемников, и на основании полученных оценок передаст таблицу  $PT_1$  и управление одному из исправных преемников (результаты этих действий показаны в соответствующих строках табл. 3). В отличие от предшественников, подсистема  $M_5$  не имеет среди своих преемников исправной подсистемы, не исполнявшей функций проверяющей. Но такая подсистема  $M_0$ , входящая в состав подсистемы  $M_{4,0}$  есть в строке  $T_4$  таблицы  $PT_1$ . Поэтому подсистема  $M_5$  передаст таблицу  $PT_1$  и управление СД подсистеме  $M_0$  через подсистему  $M_4$ . Проверив своих преемников ( $M_{0,1}$  и  $M_{0,3}$ ), подсистема  $M_0$  не найдет больше исправных подсистем, не исполнявших функций проверяющих. Следовательно, сбор тестовой информации закончен, и по цепочке:  $M_0 \rightarrow M_5 \rightarrow M_4 \rightarrow M_3 \rightarrow M_2 \rightarrow M_1$  подсистема  $M_0$  отправит таблицу  $PT_1$ , чтобы каждый из исправных процессоров получил одинаковую информацию.

Для обработки информации из таблицы  $PT_1$  исправный процессор переписывает ее в диагностическую таблицу  $T_D$  (см. табл. 2).

Обработка записей из таблицы  $PT_1$ , выполняемая каждым исправным процессором, заключается в следующем:

- из строк  $T_1, \dots, T_0$  таблицы  $PT_1$  выделяются оценки проверок  $r_{1,2} = 0, r_{1,4} = 1, \dots, r_{0,1} = 0, r_{0,3} = 0$ , и записываются в соответствующие клетки столбца  $r_{j,k}$  таблицы  $T_D$  (см. табл. 2);
- значения 0 и 1, записанные в столбце  $r_{j,k}$ , переписываются в клетки строки  $M_{j,k}$ , отмеченные знаком “\*”

Таблица 3

**Информация к Примеру**

Строка $T_j$	Проверяющая подсистема	Проверяемые подсистемы оценки их проверок			
	$M_j$	$M_{j, k1}$	$r_{j, k1}$	$M_{j, k2}$	$r_{j, k2}$
Строки с результатами контроля					
$T_0$	$M_0$	$M_{0,1}$	<b>0</b>	$M_{0,3}$	0
$T_1$	$M_1$	$M_{1,2}$	0	$M_{1,4}$	1
Таблица $PT_1$					
$T_1$	$M_1$	$M_{1,2}$	<b>0</b>	$M_{1,4}$	1
$T_2$	$M_2$	$M_{2,3}$	<b>0</b>	$M_{2,5}$	0
$T_3$	$M_3$	$M_{3,4}$	<b>0</b>	$M_{3,6}$	1
$T_4$	$M_4$	$M_{4,5}$	<b>0</b>	$M_{4,0}$	0
$T_5$	$M_5$	$M_{5,6}$	1	$M_{5,1}$	0
$T_0$	$M_0$	$M_{0,1}$	0	$M_{0,3}$	0
Конец СД					

- (так как в данном сеансе СД подсистемы  $M_{6,0}$  и  $M_{6,2}$  не проверялись, то в клетках на пересечении этих строк и столбца  $r_{j,k}$  остались прочерки, а в соответствующих клетках других столбцов — знаки “\*”);
- заполняются прочерками (—) клетки строк  $A_0, A_1, A_2$ ;
  - результаты диагностирования формируются следующим образом:
    - в  $k$ -ю клетку строки  $A_0$  записывается 0, если в вышележащих клетках этого столбца есть хотя бы один нуль;
    - в клетку на пересечении строки  $A_1$  и столбца, озаглавленного символом вершины  $M_6$ , записывается 1, так как в вышележащих клетках этого столбца есть одна единица и нет ни одного нуля;
    - в клетку на пересечении строки  $A_1$  и столбца, озаглавленного символом линии  $L(1-4)$ , записывается 1, так как вершины  $M_1$  и  $M_4$ , которые связывает эта линия, исправны (в соответствующих клетках строки  $A_0$  записаны нули);
    - в четыре клетки строки  $A_2$ ; озаглавленных символами дуг  $L(j-k)$ , записывается символ  $Z$ , так как одна из вершин ( $M_6$ ), которые связывает каждая из этих линий, подозревается в отказе (в соответствующей клетке строки  $A_1$  записана 1);
  - расшифровка записей этих строк показывает, что:
    - исправны 15 компонентов, номера которых определяются расположением нулевых записей в строке  $A_0$ ;
    - в соответствии с расположением единиц в строке  $A_1$  подозревается в отказе подсистема  $M_6$  и линия связи (1-4), что совпадает с номерами компонентов в исходной отказовой ситуации;
    - в соответствии с расположением символов  $Z$  в строке  $A_2$  остались неопределенными состояния линий (5-6), (6-0), (3-6) и (6-2), так как их проявления “маскируются” неисправным состоянием вершины 6, с которой они связаны.
- Таким образом, в данном примере идентифицированы состояния 17-ти из 21-го компонента неоднородной системы  $Q_{7,14}$ .

### ЗАКЛЮЧЕНИЕ

Представлен способ СД компонентов неоднородных цифровых систем, содержащих равное число устройств двух типов (процессоров, памяти). Для проведения процессов контроля и СД связанные разнотипные устройства объединяются в подсистемы: проверяющие и проверяемые. В основу организации СД, реализующего предложенный способ, как и в работе [4], положен принцип постепенного расширения множества исправных модулей: выполнение необходимых процедур начинается с исправной подсистемы и последовательно передается другим подсистемам, исправность которых подтверждена предшествующими проверками. Отметим, что начальная подсистема процесса СД не является

фиксированной, а определяется по результатам работы системы контроля.

Предполагается, что в исходной неоднородной системе есть диагностический монитор, который инициирует процессы контроля и СД.

В рассмотренном примере организация контроля и СД отказавших компонентов в неоднородной 14-модульной системе осуществляется с точностью до подсистемы, включающей в себя процессор, память и линию связи между ними, или линии, связывающей разные подсистемы. Требование такой точности целесообразно при диагностировании отказоустойчивых систем с реконфигурацией, в которых отказавшая подсистема автоматически заменяется исправной резервной [2, 5]. Автором был рассмотрен и другой пример организации контроля и СД отказавших компонентов в той же системе с точностью до отдельного компонента (процессора, памяти, линии связи). Для получения такой точности необходимо, чтобы проверяемые подсистемы различались номерами входящих в них устройств памяти или процессоров. Но увеличение числа различных проверяемых подсистем требует пропорционального увеличения числа необходимых проверок. В результате, снижение требуемой точности диагностирования (в приведенном примере) позволило в 3 раза уменьшить число необходимых проверок и существенно сократить трудоемкость последующих процедур. Хотя в приведенном примере показано правильное диагностирование отказов двух компонентов, но можно подобрать такие сочетания двух неисправных компонентов, диагноз которых даст неправильный результат. Причина в структуре таблицы состава, в которой каждая проверяемая подсистема тестирует по две проверяемых подсистемы, поэтому выбранные системы проверок обеспечивают достоверное диагностирование только одиночных компонентов.

### ЛИТЕРАТУРА

1. Barsi F., Grandoni F., Maestrini P. A theory of diagnosability of digital systems // IEEE Trans. Comput. — 1976. — Vol. C-25, N 6. — P. 585—593.
2. Каравай М. Ф. Минимизированное вложение произвольных гамильтоновых графов в отказоустойчивый граф и реконфигурация при отказах. Ч. 1. Одно-отказоустойчивые структуры // Автоматика и телемеханика. — 2004. — № 12. — С. 159—177.
3. Пархоменко П. П. Определение технического состояния многопроцессорных вычислительных систем путем анализа графа синдромов // Там же. — 1999. — № 5. — С. 126—135.
4. Ведешенков В. А. Организация самодиагностирования технического состояния цифровых систем // Там же. — 2003. — № 11. — С. 165—182.
5. Ведешенков В. А. Самодиагностирование отказов цифровых систем с реконфигурацией // Проблемы управления. — 2003. — № 4. — С. 39—51.

☎ (095) 334-75-90

E-mail: vva@19.ipu.rssi.ru





# ПРИМЕНЕНИЕ МЕТОДОВ РАСПОЗНАВАНИЯ ОБРАЗОВ В ЗАДАЧАХ МОЛЕКУЛЯРНОЙ БИОЛОГИИ<sup>1</sup>

А. Я. Червоненкис

*Институт проблем управления им. В. А. Трапезникова, г. Москва*

Отмечено, что новым полем применения методов распознавания образов в последнее время стали задачи молекулярной биологии — белки представляют собой последовательное соединение аминокислот, тогда как свойства молекулы ДНК определяются последовательностью нуклеотидных пар; поэтому задачи классификации белков, выделение фрагментов генома и предсказание их функций могут рассматриваться как задачи распознавания слов в заданном алфавите, которые в свою очередь оказываются подзадачами в задаче расшифровки структуры белка и генома и предсказания их функций. Предложен новый вид ядерной функции, используемой далее в машине опорных векторов для обучения распознаванию слов, и приведены результаты сравнения этого метода с другими на примере двух задач распознавания фрагментов генома. Работа выполнена в Лондонском университете (Royal Holloway University of London).

## ВВЕДЕНИЕ

В последнее время методы распознавания образов все чаще применяются для решения задач молекулярной биологии. К числу таких задач относятся, например, определение функциональных свойств белков по их структуре (первичной, вторичной или третичной), выделение в геноме отдельных участков, ответственных за управление теми или иными молекулярно-биологическими процессами.

Состав белка однозначно определяется линейной последовательностью аминокислот. Всего имеется 20 аминокислот, и поэтому линейная (первичная) структура белка задается словом в алфавите из 20 символов. Длина слова варьируется от нескольких сот до нескольких тысяч букв. Функциональные свойства белка по преимуществу определяются его вторичной и третичной структурами. Однако, последние в естественных условиях однозначно определяются первичной структурой. Таким образом, вся информация о свойствах белка закодирована словом, задающим его первичную структуру. Для ряда белков путем длительных и дорогостоящих экспериментов удается установить их функциональные свойства. Свойства новых белков, а также таких, для которых

эксперименты не проводились, приходится оценивать по сходству с теми, для которых свойства установлены экспериментально. Если сгруппировать белки в классы по их функциональным свойствам, то отнесение нового (или неисследованного) белка к тому или иному классу оказывается типичной задачей распознавания образов. При этом белки с установленными свойствами могут служить обучающей выборкой.

Аналогичные задачи возникают при исследовании генома. Свойства молекулы ДНК определяются последовательностью пар нуклеотидов. Всего (у всех организмов) встречаются четыре таких пары, которые кодируются символами *A*, *C*, *G* и *T*. Таким образом, каждая молекула ДНК может рассматриваться как слово в алфавите из четырех букв. В таком слове содержится до нескольких миллионов букв.

В настоящее время молекулы ДНК многих организмов интенсивно расшифровываются [1, 2]. После того, как экспериментально установлена последовательность нуклеотидных пар (т. е. проведено секвенирование) нужно понять, как эта молекула работает. Для этого на ней нужно, прежде всего, выделить последовательность генов — участков, содержащих коды соответствующих белков, разделенных межгенными промежутками.

При работе в организме гены сначала транскрибируются в молекулы мРНК, затем в молекулы РНК, после чего транслируются в белки. Но не весь код, транскрибируемый в РНК, служит для задания белка. Гены эукариотов содержат значительные участки, которые не транскрибируются в белок. Кодирующие зоны внутри

<sup>1</sup> Работа доложена на Научных чтениях памяти профессора А. М. Петровского, Москва, Ин-т проблем управления, 17 марта 2005 г.

гена называют *эксонами*, а некодирующие — *интронами*. Соответствующие белковые комплексы “распознают” начало и конец интрона и исключают его при очередном копировании. Этот процесс называется *сплайсингом*. После того, как из слова, кодирующего ген, исключены интроны (а также участок между началом транскрипции и началом трансляции), по нему можно прочитать первичную структуру белка, закодированного в гене. Известно, что каждой аминокислоте соответствует триплет нуклеотидных пар (слово из трех букв в алфавите *A, C, G* и *T*). Таким образом, задача выделения кодирующей части гена, т. е. разделение его на экзоны и интроны, оказывается весьма актуальной.

В то же время по известным белкам, порождаемым данной группой генов, можно решить обратную задачу — указать экзоны, кодирующие эти белки. Случаи с известным разделением на экзоны и интроны можно использовать в качестве обучающей последовательности в задаче обучения распознаванию точек перехода от эксона к интрону и наоборот.

Гену предшествует (или частично захватывает его начало) так называемая промоуторная зона, содержащая промоуторы — участки ДНК, к которым присоединяются белковые комплексы, управляющие работой гена. Они включают или выключают, усиливают или ослабляют экспрессию гена. Поэтому задача распознавания промоуторов также оказывается весьма актуальной. Белковые комплексы своими средствами “распознают” соответствующие промоуторы, и, значит, эта задача должна быть разрешима и машинными средствами. В ряде случаев положение промоуторов установлено экспериментально. Эти примеры могут служить обучающей выборкой.

Существует много других участков (сайтов), регулирующих функционирование генома, по отношению к которым можно поставить аналогичные задачи распознавания. Разумеется, при расшифровке генома все эти задачи должны решаться в комплексе. Но распознавание отдельных участков может рассматриваться как система элементарных блоков (кирпичей) при решении общей задачи.

Различные методы обучения распознаванию образов связаны с соответствующей метрикой, определяющей степень близости объектов. Это может быть простая евклидова метрика, метрика, задаваемая положительно определенной квадратичной формой или ядерной функцией, а также иные метрики. В методах, основанных на построении разделяющей поверхности, метрика используется неявно — через соответствующую форму скалярного произведения, а в методе ближайшего соседа — явно. В последнее время появилось (и появляется) все больше работ по применению методов распознавания образов в молекулярной биологии с применением всех указанных средств [1–4].

Применительно к текстовым строкам (при кодировании строки вектором) простая Евклидова метрика соответствует числу несовпадающих букв в двух словах; метрика, задаваемая квадратичной формой, — степени различия с учетом различных штрафов, зависящих от конкретных букв и их положения в слове, а ядерные функции — более сложным схемам сравнения двух строк.

Близкие по функциям белки, как правило, произошли в ходе эволюции от одного прародителя: одни аминокислоты заменялись на другие, некоторые звенья

выпадали, вставлялись новые. Возможна также перестановка, вставка или удаление больших фрагментов. Сказанное относится также к фрагментам генома, выполняющим сходные функции, поскольку они “распознаются” одинаковыми или сходными белковыми комплексами.

Если бы дело ограничивалось только заменами, то близость слов, кодирующих белки (или фрагменты генома), можно было бы мерить числом несовпадений или штрафной функцией с учетом различных замен. Но при наличии вставок и выпадений нужна более сложная процедура. Дело в том, что выпадение хотя бы одной буквы сдвигает остаток слова, и, даже если он полностью сохраняется, число совпадений будет мало. В связи с этим уже давно разработан так называемый метод выравнивания [3] (*alignement*). Назначается матрица штрафов и штраф (или штрафы) за пропуск буквы в одном из слов. Далее методом динамического программирования ищется такая последовательность пропусков и соответствий, которая обеспечивает минимум суммы штрафов. Метод, основанный на этих идеях, реализован в стандартной программе BLAST [4], используемой для классификации белков и других молекулярных цепей.

## 1. ЯДРО ВЫРАВНИВАНИЯ

Функция штрафа может интерпретироваться в вероятностных терминах. Представим себе, что некоторый генератор порождает одновременно два слова. Каждый такт генерируется или одновременно по одной букве в каждом слове с вероятностью  $P(a, b)$ , где  $a$  и  $b$  — перемешанные, пробегающие алфавит,  $P(a, b) = P(b, a)$ , или в одном слове генерируется буква, а в другом — пробел с вероятностью  $\pi(a)$ . После этого пробелы убираются и получаются два слова  $x$  и  $y$ . Зная только результат, можно по-разному расставлять пробелы в полученные слова. Для каждой расстановки пробелов можно подсчитать вероятность получения такой комбинации: она будет равна произведению вероятностей порождения соответствующих пар (буква — буква, буква — пробел, пробел — буква). Если за штраф, соответствующий паре, принять  $S(a, b) = -\log P(a, b)$ ,  $S(a, \_) = S(\_, a) = -\log \pi(a)$ , то суммарный штраф будет равен взятому со знаком минус логарифму вероятности данной расстановки пробелов в словах  $x$  и  $y$ .

Обычный метод выравнивания соответствует поиску наиболее вероятной расстановки пробелов. Чем выше эта вероятность, тем ближе слова. Однако более естественно в качестве меры близости взять не вероятность наиболее правдоподобной расстановки пробелов, а вероятность  $P(x, y)$  одновременного порождения слов  $x$  и  $y$ . Она будет равна сумме вероятностей всех вариантов расстановки пробелов. С. Watkins [5] предложил использовать вероятность  $P(x, y)$  в качестве ядра в методах распознавания. В той же работе показано, что такая ядерная функция будет положительно определенной.

Однако полный перебор всех вариантов расстановки пробелов требует комбинаторно большого числа шагов. Нами был предложен быстрый вариант вычисления вероятности  $P(x, y)$ , который строится по схеме, близкой к динамическому программированию [1]. Кроме того, непосредственное использование вероятности  $P(x, y)$  в качестве ядра имеет ряд недостатков. Во-первых, эта величина очень резко зависит от длины слов, что существ-



венно при сравнении слов разной длины. Во-вторых, она очень быстро убывает при уже небольших различиях в словах. Поэтому, чтобы привести ядро к виду, удобному для применения, предложено провести следующие операции:

- нормировка:  $K^*(x, y) = (P(x, y))^{1/l}$ ,  $K^{**}(x, y) = K^*(x, y) / (K^*(x, x) K^*(y, y))^{1/2}$ ;
- расширение:  $K(x, y) = (K^*(x, y))^q$ , где  $l$  — длина слова,  $q$  ( $0 < q \leq 1$ ) — параметр ядра.

Изменяя параметр  $q$ , можно “расширять” или “сужать” ядро и соответственно менять меру близости между словами.

Предлагаемое ядро (ядерная функция) получило название *ядра выравнивания* (alignment kernel). Это ядро предлагается использовать в рамках машины опорных векторов (SVM) [6], но его можно применить и как меру близости в алгоритме ближайшего соседа.

С применением этого ядра и машины опорных векторов решались две задачи: распознавание промоторов  $\sigma^{70}$  у бактерии *E-coli* (*Escherichia coli*, кишечная палочка) и распознавание переходов “экзон—интрон” и “интрон—экзон” (сайтов сплайсинга) у растения *Arabidopsis thaliana*. Эти виды были выбраны потому, что их геномы полностью секвенированы (экспериментально установлена последовательность нуклеотидных пар) и достаточно аннотированы, чтобы составить обучающую и тестовую выборки. Предлагаемый метод сравнивался на том же материале с другими, как использующими SVM (но с другими ядрами), так и не опирающимися на SVM.

## 2. РАСПОЗНАВАНИЕ ПРОМОУТОРА $\sigma^{70}$ У БАКТЕРИИ *E-COLI*

Бактерия *E-coli* (прокариот) имеет одну кольцевую хромосому, которая в настоящее время полностью секвенирована, т. е. у нее полностью установлена последовательность нуклеотидных пар. Иными словами, для нее найдено кольцевое слово, кодирующее эту последовательность. В ней экспериментально или по косвенным признакам выделено приблизительно 4000 генов. Работа примерно 80 % из них регулируется белковым комплексом (РНК-полимеразой), содержащим белок  $\sigma^{70}$ . Этот белок присоединяет РНК-полимеразу к соответствующему промотору (участку ДНК), предшествующему гену, называемому  $\sigma^{70}$ -промотором. После этого РНК-полимераза запускает транскрипцию гена. Гены, запускаемые  $\sigma^{70}$ -промотором, ответственны за рост бактерии. Считается, что положение  $\sigma^{70}$ -промотора тесно связано с точкой начала транскрипции и может служить для ее определения. Иногда один промотор запускает транскрипцию сразу целой цепочки генов. Такая цепочка называется опероном. В то же время некоторые гены внутри оперона могут запускаться своим промотором.

Мы собрали вместе список экспериментально установленных точек начала транскрипции (ТНТ), соединив данные из RegulonDB [7] и PromEC [8]. За промоторную зону принимался участок, содержащий 60 нуклеотидных пар (букв слова), предшествующих ТНТ, и 19 пар, следующих за ней. Считается, что  $\sigma^{70}$ -промотор заведомо находится внутри этой зоны.

Задача состояла в том, чтобы научиться отличать такие участки от случайно выбранного участка ДНК. В качестве положительных примеров (как для обучения, так и для теста) использовались упомянутые участки, содержащие экспериментально установленную точку начала транскрипции и  $\sigma^{70}$ -промотор. В качестве отрицательных — случайно выбранные участки хромосомы той же длины, при условном предположении, что ТНТ в них, как и в положительных примерах, расположена в 61-й позиции слова. Поскольку частотный состав нуклеотидных пар различен в кодирующей и не кодирующей частях ДНК, было сформировано два блока отрицательных примеров — один из участков, отобранных в кодирующей части, другой — из межгенных промежутков. Всего было использовано 699 слов по 80 букв в качестве положительных примеров, 709 слов из кодирующей и 709 слов из не кодирующей частей как отрицательные примеры.

Для практического применения эта задача должна решаться, конечно, в комбинации с другими задачами (выделение кодирующей части гена, поиск точки начала трансляции, определение промоторов другого типа и др.). Но для сравнения различных методов интересно было исследовать “чистую” задачу распознавания двух классов слов.

Для обучения и проверки использовались все положительные примеры и одна из двух групп отрицательных примеров. Затем случайно отбиралась половина примеров на обучение и столько же на экзамен. Такое случайное разбиение повторялось 100 раз, и результаты были усреднены по всем 100 экспериментам.

Сравниваемые методы обучения можно разделить на две группы: универсальные методы и методы, использующие специальные свойства расположения  $\sigma^{70}$ -промотора относительно ТНТ, о которых будет сказано ниже.

В первую группу были, прежде всего, включены применение SVM с описанным выше ядром выравнивания (ЯВ + SVM) и алгоритм, использованный в стандартной программе BLAST. Последняя вычисляет меру близости двух слов, исходя из наиболее вероятной расстановки пробелов. Новое слово сравнивается со всеми словами из обучающей выборки, находится ближайшее и выясняется, к какому классу оно было отнесено по данным обучения. Новое слово относится к этому же классу. Оба этих метода учитывают, что при сравнении двух слов нужно принимать во внимание возможные вставки и пропуски, но ЯВ учитывает суммарную вероятность всех возможных вариантов расстановки пробелов, тогда как программа BLAST учитывает только наиболее вероятную из них.

Еще один метод основан на том, что частоты встречаемости отдельных букв и их комбинаций по две, три и более существенно отличаются в промоторной зоне и в случайно выбранном участке ДНК [9]. Более того, эти частоты оказываются разными в зависимости от расстояния до точки начала транскрипции. Обозначим этот метод как Зоны + SVM. Слово разбивается на пять зон. В каждой из этих зон по обучающей выборке подсчитываются частоты встречаемости отдельных букв и их сочетаний. Логарифм частоты принимается за оценку буквы или сочетания. Для произвольного слова вычисляется его “правдоподобие” по зонам (т. е. для каждой зоны отдельно) путем сложения оценок правдоподобия действительно встретившихся там букв или сочетаний. По-

лученные пять оценок принимаются за координаты вектора, характеризующего слово. Далее ядро вычисляется как простое скалярное произведение этих векторов и используется в SVM. Этот метод основан не на прямом сравнении двух слов, а на сравнении частот встречаемости букв и их сочетаний.

Два метода из этой группы основаны на прямом применении машины опорных векторов (SVM) без учета вставок и удалений. Слово кодируется вектором путем замены каждой буквы на четыре бинарных признака:

$$\begin{aligned} A &\rightarrow (0, 0, 0, 1), \\ C &\rightarrow (0, 0, 1, 0), \\ G &\rightarrow (0, 1, 0, 0), \\ T &\rightarrow (1, 0, 0, 0). \end{aligned}$$

Алгоритм “простое ядро + SVM” (ПЯ + SVM) использует далее полиномиальное ядро вида  $(1 + N)^d$ , где  $N$  — число совпадающих букв в двух сравниваемых словах,  $d$  — параметр алгоритма. Наилучшим оказалось значение  $d = 3$ .

Алгоритм “локально исправленное ядро” (ЛИЯ + SVM) использует более сложное ядро [10]. Два “окна” бегут параллельно вдоль двух сравниваемых слов. При каждом положении окон попавшее в них содержимое сравнивается с помощью полиномиального ядра, указанного выше. Далее сумма полученных значений (по всем положениям окна) возводится в заданную степень, и результат считается значением ядра. Параметры алгоритма — ширина окна, степень внутреннего полинома, степень, в которую возводится сумма — подбирались.

**Другая группа методов** основана на специальных свойствах промоуторной зоны, содержащей  $\sigma^{70}$  промоутор. В работах генетиков [10–13] установлено, что на расстояниях  $-10$  и  $-35$  нуклеотидных пар от точки начала транскрипции находятся два “консервативных” блока, называемых  $-10$  и  $-35$  боксами. “Консервативный” означает, что в них обычно повторяется одна и та же последовательность букв — TATAAT для  $-10$  бокса и TTGAC для  $-35$  бокса. Однако оказалось, что очень часто те или иные буквы заменяются на другие и положение боксов относительно ТНТ может “плавать”. Можно составить матрицы, характеризующие вероятность замены буквы на другую в зависимости от ее положения в боксе и кандидата на замену [14, 15]. Если бы эти матрицы и распределение вероятностей возможных сдвигов боксов от их стандартного положения были известны, то в предположении независимости признаков можно было бы вычислить функцию правдоподобия и сравнить ее с такой же функцией для случайно выбранного участка. Но имеющиеся в литературе сведения (для *E-coli*) основаны на меньшем объеме данных, нежели тот, которым располагали мы. Поэтому оценку матрицы вероятностей замены и распределения вероятностей сдвигов мы включили в процесс обучения. Для этого был применен прием максимизации правдоподобия, предложенный в работе [16]. Он состоит в следующем. Задаются некоторой матрицей штрафов (например, известной из литературы) и распределением вероятностей сдвигов боксов (например, равномерным в заданном диапазоне). По этим оценкам находится наиболее правдоподобное положение боксов на всех положительных объектах обучающей выборки. При этом положении подсчитываются частоты встречаемости букв во всех положениях внутри боксов и фактическое распределение вероятности

сдвигов. Затем операция повторяется с новой матрицей и новым распределением, пока при очередной итерации боксы во всех примерах не сохранят свое положение. Полученные в результате распределения принимаются за оценку истинных. По ним для произвольного слова вычисляется значение правдоподобия того, что оно действительно содержит промоутор: ищется наиболее правдоподобное положение боксов и перемножаются вероятности букв в заданных позициях и вероятности выбранных сдвигов (или складываются их логарифмы). Порог распознавания находится путем сравнения значений правдоподобия для положительных и отрицательных примеров обучающей выборки. Для распознавания нового объекта вычисляется его оценка правдоподобия и сравнивается с порогом. Этот метод мы обозначаем как “боксы + максимум правдоподобия” (Боксы + МП).

Сложение логарифмов вероятности при вычислении правдоподобия основано на гипотезе о независимости факторов, что, быть может, не соответствует действительности. Поэтому в качестве альтернативы был испытан более сложный прием (Боксы + МП + SVM). Матрицы и распределение вероятностей сдвигов находятся так же, как и в предыдущем методе. Так же находится наиболее правдоподобное положение сдвигов. Но далее логарифмы вероятностей не складываются, а служат координатами вектора, кодирующего слово. Полученные векторы обрабатываются затем машиной опорных векторов с простым ядром.

Более прямой метод применения SVM после того, как найдено положение боксов (Боксы + SVM) состоит в следующем. Как и раньше, оцениваются матрицы замены и вероятности сдвигов боксов. Далее для произвольного слова находится наиболее правдоподобное положение боксов, и их содержимое объединяется в новое слово. Это слово кодируется бинарным вектором, как указано выше, и используется в SVM с простым полиномиальным ядром.

В литературе имеются указания, что на сцепление белка  $\sigma^{70}$  с геномом помимо  $-10$  и  $-35$  боксов влияют также и другие регуляторные сайты (участки), встречающиеся в промоуторной зоне. В качестве таковых нами были использованы типовые последовательности:

INF: ‘WATCAANNNTTR’ (Friedman 1988),

LexA: ‘TACTGTATATATACAGTA’ (Hertz et al. 1990),

CRP: ‘AAATGTGATCTAGATCACATTT’ (Wang and Church 1992).

Характерное положение этих сайтов относительно ТНТ не установлено, многие буквы могут заменяться на другие, а случаев, где такие сайты наблюдались экспериментально, мало. Тем не менее, поскольку учет этих участков дает дополнительную информацию, мы испытали схему, близкую к методу Боксы + МП + SVM, обозначив ее “Боксы + Регуляторные сайты + Максимум правдоподобия + SVM” (Боксы + РС + МП + SVM).

В приведенных далее таблицах качество распознавания оценивалось по следующим показателям:

- общее число ошибок (ОШ) — процент ошибочно распознанных примеров среди всех объектов, предъявленных на экзамене;
- пропуск цели (ПЦ) — процент ошибок среди положительных примеров на экзамене;





Таблица 1

**Результаты экзамена, усредненные по 100 вариантам разбиения на обучение и тест. Отрицательные примеры из кодирующей части**

Метод	ОШ	ПЦ	ЛТ
ЯВ + SVM	<b>16,5</b>	<b>18,5</b>	14,6
BLAST	34,6	40,9	28,7
Зоны + SVM	21,0	32,2	<b>10,4</b>
ПЯ + SVM	29,1	29,8	28,4
ЛИЯ + SVM	19,3	24,9	14,1
Боксы + МП	19,5	24,4	14,8
Боксы + МП + SVM	19,1	23,6	14,8
Боксы + SVM	21,6	23,2	20,0
Боксы + PC + МП + SVM	16,8	22,7	11,3

Таблица 2

**Результаты экзамена, усредненные по 100 вариантам разбиения на обучение и тест. Отрицательные примеры из некодирующей части**

Метод	ОШ	ПЦ	ЛТ
ЯВ + SVM	18,6	19,0	18,2
BLAST	35,4	40,9	30,2
Зоны + SVM	22,5	33,1	12,5
ПЯ + SVM	33,5	35,4	31,7
ЛИЯ + SVM	23,5	38,8	9,1
Боксы + МП	21,0	28,4	14,0
Боксы + МП + SVM	20,5	25,6	15,7
Боксы + SVM	23,0	23,8	22,2
Боксы + PC + МП + SVM	30,3	25,7	34,6

- ложные тревоги (ЛТ) — процент ошибок среди отрицательных примеров на экзамене.

При сравнении результатов следует помнить, что путем изменения порога ошибки типа “пропуск цели” легко переходят в ошибки типа “ложная тревога”. По суммарному числу ошибок, как видно из табл. 1 и 2, предложенный нами метод ЯВ + SVM имеет преимущество перед всеми другими сравниваемыми методами.

### 3. РАСПОЗНАВАНИЕ ТОЧЕК ПЕРЕХОДА “ЭКСОН—ИНТРОН” И “ИНТРОН—ЭКСОН” У *ARABIDOPSIS THALIANA*

*Arabidopsis thaliana* — это растение (эукариот), обладающее очень коротким циклом размножения. По этой причине оно интенсивно используется в генетических экспериментах. Геном *Arabidopsis thaliana*, в настоящее время полностью секвенированный, состоит из пяти хромосом и имеет общую длину более 117 миллионов нуклеотидных пар. На текущий момент в этом геноме выделено 4714 генов, расшифрованных “Ceres. Inc.”. Только у 20 % генов кодирующая часть непрерывна. У 3703 генов кодирующая часть (эксоны) прерывается некодирующими участками (интронами). Важнейший момент в расшифровке генома состоит в том, чтобы найти точки, где заканчивается эксон и начинается интрон (доноры), и, наоборот, точки, где заканчивается интрон и начинается очередной эксон (акцепторы). Схема распознавания этих точек, установленная для *Arabidopsis thaliana*, видимо, может быть применена и для других растений.

У всех выделенных генов экспериментально (или по косвенным признакам) установлено деление на интроны и экзоны и, соответственно, найдены точки перехода. Все эти данные использовались как положительные примеры в обучающей и тестовой выборках. Более 99 % процентов интронов начинается с пары букв GT и в более чем 99 % случаев перехода от интрона к эксону интрон заканчивается парой букв AG. Конечно, эти признаки не достаточны для распознавания, так как такие двухбуквенные сочетания встречаются достаточно часто в произвольных частях генома. Но поиск точек перехода может быть ограничен этим условием. Поэтому в качестве отрицательных примеров брались случайно выбранные участки ДНК, центрированные вокруг пары букв GT (для доноров) или пары AG (для акцепторов).

Поскольку эксон служит для кодирования белка, то информация о точке перехода должна быть сосредоточена в основном на участке интрона, прилегающем к этой точке. Поэтому для распознавания доноров в качестве положительных примеров брались участки ДНК, включающие в себя 30 нуклеотидных пар (букв) из интрона и 10 пар из эксона, прилегающих к точке перехода. Для акцепторов — 10 пар из эксона и 50 пар из интрона. В качестве отрицательных примеров брались слова аналогичного формата, центрированные соответственно относительно случайно выбранных пар букв GT (для доноров) или AG (для акцепторов).

Всего для обучения распознаванию (как доноров, так и акцепторов) и для экзамена использовались 2000 положительных примеров и 8000 отрицательных примеров на обучение, по 5000 положительных и отрицательных примеров на экзамен.

Буквы (нуклеотидные пары) кодировались, как и в предыдущей задаче, четырьмя бинарными признаками:

$$\begin{aligned} A &\rightarrow (1, 0, 0, 0), \\ C &\rightarrow (0, 1, 0, 0), \\ G &\rightarrow (0, 0, 1, 0), \\ T &\rightarrow (0, 0, 0, 1). \end{aligned}$$

При практическом применении распознавания необходимо учесть, что пары букв GT и AG встречаются в ДНК гораздо чаще, чем фактические точки перехода. С другой стороны, имеется возможность дополнительной проверки правильности распознавания: выделенная в виде эксонов часть гена должна быть сплошным кодом белка, т. е. не должна содержать некодирующих триплетов. Но это выходит за рамки настоящего исследования. Для сравнения различных методов собственно распознавания предлагаемая постановка достаточно удобна.

В качестве фонового метода рассматривалось сравнение правдоподобия гипотезы 1 о том, что данный пример относится к числу положительных, против гипотезы 2, что он относится к классу отрицательных. В этой задаче не известно характерных участков (сайтов). Поэтому просто оценивалась (по частоте встречаемости в обучающей выборке) вероятность  $P(a, n)$  встретить букву  $a$  в  $n$ -й позиции слова для положительных примеров и такая же вероятность  $P^*(a, n)$  — для отрицательных. Для нового слова в предположении независимости правдоподобие гипотезы 1 вычислялась как сумма логарифмов  $P(a, n)$  по всем позициям слова, где  $a$  — буква, фактически встретившаяся в  $n$ -й позиции, и правдоподобие гипотезы 2 как сумма логарифмов  $P^*(a, n)$ . За истинную принималась та гипотеза, у которой прав-

доподобие больше. Этот метод известен как ПЗВМ (позиционно-зависимая весовая матрица).

В табл. 3 приведены результаты этого метода на материале теста.

Этот метод сравнивался с результатами распознавания с помощью машины опорных векторов (SVM) при различных ядрах, а также методом ближайшего соседа в евклидовой метрике (БСЕ) и на основе стандартной программы BLAST.

В качестве ядерных функций использовались описанное ранее ядро выравнивания при наилучшем значении параметров (ЯВ + SVM), локально исправленное ядро при наилучшем значении параметров (ЛИЯ + SVM), простое скалярное произведение  $(\mathbf{x}, \mathbf{y})$  (СП + SVM), полиномиальное ядро  $((\mathbf{x}, \mathbf{y}) + 1)^d$  при различных степенях  $d$  (ПЯ + SVM) и Гауссово ядро (ГЯ + SVM) —  $\exp(-q |\mathbf{x} - \mathbf{y}|^2)$  при различных значениях параметра  $q$ .

Результаты распознавания приведены в табл. 4.

Наилучший результат получен для локально исправленного ядра. Заметим, что все методы, использующие SVM (при нормальном выборе параметров), дали лучший результат, чем другие. Ядро выравнивания на этот раз не дало преимущества. Отметим также, что применение простого скалярного произведения в качестве ядра (т. е. обычный обобщенный портрет) дало результат, близкий к оптимальному. Решающее правило в этом случае может быть записано в форме весовой матрицы. Однако, в отличие от весовых матриц, которые строятся

Таблица 3

**Результаты распознавания точек перехода с помощью ПЗВМ**

Вид перехода	ОШ	ПЦ	ЛТ
Доноры	7,92	12,84	3,00
Акцепторы	12,73	20,36	5,10

Таблица 4

**Результаты экзамена**

Метод	Параметр	ОШ	ПЦ	ЛТ
ЯВ + SVM	—	7,41	12,84	1,96
ЛИЯ + SVM	—	5,99	10,04	1,94
СП + SVM	—	6,88	11,62	2,14
ПЯ + SVM	$d = 2$	6,44	10,68	2,20
	$d = 3$	6,56	11,00	2,02
	$d = 4$	7,05	12,38	1,72
	$d = 5$	8,19	15,24	1,14
ГЯ + SVM	$d = 0,001$	6,81	11,52	2,10
	$d = 0,01$	6,62	11,26	1,98
	$d = 0,025$	6,51	11,00	2,02
	$d = 0,04$	6,77	11,70	1,84
БСЕ	—	22,24	29,52	14,96
BLAST	—	37,8	63,3	12,3

по всему материалу обучения, эта матрица построена только по крайним (опорным) векторам со специально подобранными весами, т. е. по наиболее трудным для распознавания примерам. Это существенно в тех случаях, когда распознавание “трудных” объектов требует иного подхода, чем в среднем.

**ЗАКЛЮЧЕНИЕ**

Полученные результаты свидетельствуют о высокой эффективности методов распознавания, основанных на применении машины опорных векторов. Дальнейшая работа будет направлена на включение элементов распознавания в более сложные системы, предназначенные для анализа структуры белка и генома.

**ЛИТЕРАТУРА**

1. *Sequence Alignment Kernel for recognition of promoter regions* / Л. Гордон, А. Червоненкис, А. Гаммерман, И. Шахмурадов, В. Соловьев // *Bioinformatics*. — 2003. Vol. 20. — P. 1964—1971.
2. *Genom-wide prokariotic promoter recognition based on Sequence Alignment Kernel* / Л. Гордон, А. Червоненкис, А. Гаммерман, И. Шахмурадов, В. Соловьев // Труды конференции IDA2003. Берлин, 2003. В кн. *Advances in Intelligent Data Analysis*. — Springer-Verlag. Vol. LNCS 2810. — P. 386—396.
3. *Goth O.* An improved algorithm for matching biological sequences // *J. Mol. Bio.* — 1982. — 162 (3). — P. 705—708.
4. *Gapped BLAST and PSI-BLAST: a new generation of protein search programs* / S. Altschul, T. Madden, A. Schaffer, et al. // *Nucleic Acids Res.* — 1997. — Vol. 25. — P. 3389—3402.
5. *Watkins C.* Dynamic alignment kernels // *Advances in Large Margin Classifiers*. MA: MIT Press, 2000. Cambridge, P. 39—50.
6. *Vapnik V.* Statistical learning theory. — New-York: Wiley, 1998.
7. *Regulondb* (version 3.0): transcriptional regulation and operon organization in *Escherichia coli* K-12 / H. Salgado, A. Santos-Zavaleta, S. Gama-Gastro, et al. // *Nucleic Acids Res.*, — 2000. — Vol. 28 (1). P. 65, 66.
8. *Promec: An updated database of Escherichia coli mRNA promoters with experimentally identified transcriptional start sites* / R. Hershberg, G. Bejerano, A. Santos-Zavaleta, H. Margalit // *Nucleic Acids Res.* — 2001. 29 (1). — P. 277.
9. *Oppon J., Hide W.* A statistical model for prokariotic promoter prediction // In the Nineteenth Workshop on genomic Informatics. — P. 271—273. Poster.
10. *Sholkopf D., Gederger J., Lapalme G.* Frequency of insertion/deletion, transversion and transition in evolution of 5S ribosomal RNA // *Journal of Molecular Evolution*. — 1996. — N 7. — P. 133—149.
11. *Prinbow D.* Nucleotide sequence of an rna polymerase binding site at an early t7 promoter // *Proc. of Natl Acad Sci USA*. 1975. — Vol. 72. P. 784—788.
12. *Shaller H., Gray C., Herrmann K.* Nucleotide sequence of an rna polymerase binding site from dna of bacteriophage fd // *Proceedings of Natl Acad Sci USA*. Vol. 72. P. 737—741.
13. *Sequence of promoter for coat protein gene of bacteriophage fd* / K. Takanami., K. Sigimoto, H. Sugisaki., T. Okamoto // *Nature*. — 1976. Vol. 260 (5549). — P. 297—302.
14. *Staden R.* Computer methods to locate signals in nucleic acid sequences // *Nucleic Acids Res.* — 1984. Vol. 12 (1, pt 2). — P. 502—519.
15. *Harley C., Reynolds R.* Analysis of *E. coli* promoter sequences // *Nucleic Acids Res.* — 1987. Vol. 15 (5). — P. 2343—2361.
16. *Bailey T., Elkan C.* Unsupervised learning of multiple motives in biopolymers using expectation maximization // *Machine learning* 21 (1—2). P. 51—80.

☎ (095) 334-88-20

E-mail: chervnks@ipu.rssi.ru





# МОДЕЛИРОВАНИЕ ЭФФЕКТИВНОСТИ ДЕЙСТВИЯ ПРОТИВООПУХОЛЕВЫХ ПРЕПАРАТОВ В СВЕРХМАЛЫХ ДОЗАХ ДЛЯ ОПТИМИЗАЦИИ РЕЖИМОВ ИХ ВВЕДЕНИЯ<sup>1</sup>

Н. А. Бабушкина<sup>(1)</sup>, Л. А. Островская<sup>(2)</sup>, В. А. Рыкова<sup>(2)</sup>, М. М. Фомина<sup>(2)</sup>,  
Н. В. Блюхтерова<sup>(2)</sup>, Е. Б. Бурлакова<sup>(2)</sup>, А. В. Кулешова<sup>(3)</sup>

<sup>(1)</sup> *Институт проблем управления им. В.А. Трапезникова, г. Москва;*

<sup>(2)</sup> *Институт биохимической физики им. Н.М. Эмануэля, г. Москва;*

<sup>(3)</sup> *Московский институт радиотехники, электроники и автоматики*

Методом математического моделирования проведена оценка эффективности применения противоопухолевого препарата — доксорубина — в терапевтической и сверхмалых дозах в отношении экспериментальной опухоли мышей (карциномы Льюис). Получена бимодальная зависимость “доза—эффект”. Показано, что введение сверхмалых доз цитостатика приводит к изменению параметров модели роста опухоли относительно контроля. Рассмотрены различные подходы к объяснению механизмов действия лекарственных препаратов в сверхмалых дозах.

## ВВЕДЕНИЕ

Лечение любого патологического состояния требует не только подбора адекватного препарата или комбинации лекарственных средств, но и разработки оптимальной схемы их применения. Для управления этим процессом необходимо количественно оценить ответную реакцию объекта лечения на введение препарата, выражаемую обычно в виде зависимости “доза—эффект”. В качестве дозы можно рассматривать любое количественно определенное воздействие в соответствии с применяемым способом лечения. Показателями эффекта могут служить различные характеристики, отражающие изменение состояния объекта в процессе лечения.

Противоопухолевые препараты, как правило, применяются в области доз, близких к максимально переносимым. Действие препаратов в этом диапазоне доз характеризуется пропорциональной зависимостью эффекта от дозы. Необходимость “высокодозовой” терапии обусловлена недостаточной избирательностью действия

большинства современных цитостатиков, которые в стандартных терапевтических дозах наряду с поражающим действием на опухоль вызывают токсические нарушения в различных системах организма и приводят к развитию нежелательных побочных явлений.

В литературе накоплен обширный фактический материал, свидетельствующий о биологических эффектах лекарственных препаратов в сверхмалых дозах [1–14]. Проблема изучения сверхмалых доз (СМД) лекарственных соединений, в первую очередь, противоопухолевых препаратов, заслуживает особого внимания как с точки зрения исследования фундаментальных механизмов их действия, так и для разработки новых научно-обоснованных подходов к оптимизации терапевтических воздействий.

На протяжении ряда лет в Институте биохимической физики им. Н. М. Эмануэля РАН ведутся исследования данной проблемы. Экспериментально доказана биологическая активность ряда цитостатиков — нитрозометилмочевины, циклофосфана, доксорубина — при их применении в диапазоне СМД  $10^{-5}$ – $10^{-20}$  М (1 М = 1 моль/л) в отношении экспериментальной опухолевой модели — карциномы легких Льюис [12–14]. Установлено, что направленность и выраженность биологического эффекта цитостатиков в области СМД зависит как от природы

<sup>1</sup> Работа доложена на Научных чтениях памяти профессора А. М. Петровского, Москва, Ин-т проблем управления, 17 марта 2005 г.

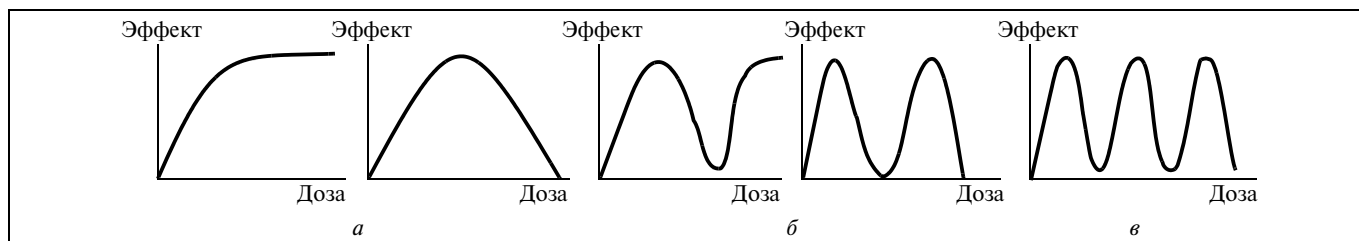


Рис. 1. Виды зависимостей “доза — эффект” для различных препаратов и биологически активных веществ в сверхмалых дозах: *a* — мономодальные; *b* — бимодальные; *e* — полимодальная

действующего агента, так и от дозы препарата и штамма опухоли. Обнаружена способность цитостатиков (нитрозометилмочевина, доксорубин) оказывать в СМД противоопухолевый эффект, сопоставимый с активностью препаратов в стандартных дозах ( $10^{-2}$ – $10^{-3}$  М) [12–14].

Однако, как показали многочисленные эксперименты [1], при использовании препаратов в области сверхмалых доз характер дозовой зависимости принципиально меняется, приобретает вид так называемых “полимодальных” кривых (рис. 1), и получаемый эффект при равномерном снижении дозы может меняться от положительного до нулевого или даже отрицательного эффекта, выражающегося в стимуляции роста опухоли. Наличие столь сложных “полимодальных” дозовых зависимостей требует большой точности в определении эффективной дозы, так как даже незначительное отклонение от нее может привести к прямо противоположному эффекту либо к отсутствию эффекта. До настоящего времени нет общепринятой теории для описания механизмов, лежащих в основе действия СМД, которая позволила бы объяснить и построить “полимодальные” дозовые зависимости. Поэтому они строятся экспериментальным путем. В результате проводимых исследований накоплен обширный экспериментальный материал в виде кинетических кривых изменения размеров опухоли в отсутствие воздействия и после него для различных режимов химиотерапии [12–14]. Этот материал позволяет применить метод математического моделирования для описания роста опухоли и получения параметров, которые могут служить критерием оценки результата воздействия препарата на опухоль, и для построения зависимости эффекта от дозы.

Применительно к онкологии практическая направленность такого рода исследований состоит в отходе от концепции “высокодозовой” терапии и попытке разработки методов применения известных цитостатиков в сверхмалых дозировках, способных обеспечить ингибирование опухолевого роста без развития токсических побочных явлений.

## 1. ОПИСАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ РОСТА ОПУХОЛИ

В общем виде процесс роста опухоли описывался дифференциальным уравнением:

$$\partial X/\partial t = \lambda(t) \cdot X(t), \quad (1)$$

где  $X(t)$  — размер опухоли,  $t$  — время,  $\lambda(t)$  — параметр, характеризующий скорость роста опухоли.

Выбор функции  $\lambda(t)$  определялся по экспериментальным кривым роста опухоли методом регрессионного анализа в системе MATLAB 6.1. Кривые роста карциномы Льюис имеют S-образной вид с насыщением и для их описания используется функция Гомпертца, которая и была принята в качестве модели [15–17]:

$$X(t) = X_0 \cdot \exp(\alpha[1 - \exp(-\beta t)]), \quad (2)$$

где  $X_0$  — начальный размер опухоли,  $\alpha$  и  $\beta$  — параметры роста опухоли.

Это уравнение получается путем решения уравнения (1) при  $\lambda(t) = \alpha\beta \cdot \exp(-\beta t)$ . Точность приближения расчетных и экспериментальных кривых оценивалась по значению среднего квадратичного отклонения (с. к. о.) методом наименьших квадратов (см. таблицу).

Как известно, введение препарата в терапевтических дозах приводит к гибели определённой части популяции опухолевых клеток, что вызывает временное уменьшение размеров опухоли, после чего ее рост возобновляется. В связи с тем, что погибшие клетки выводятся из организма в течение нескольких суток после их гибели,

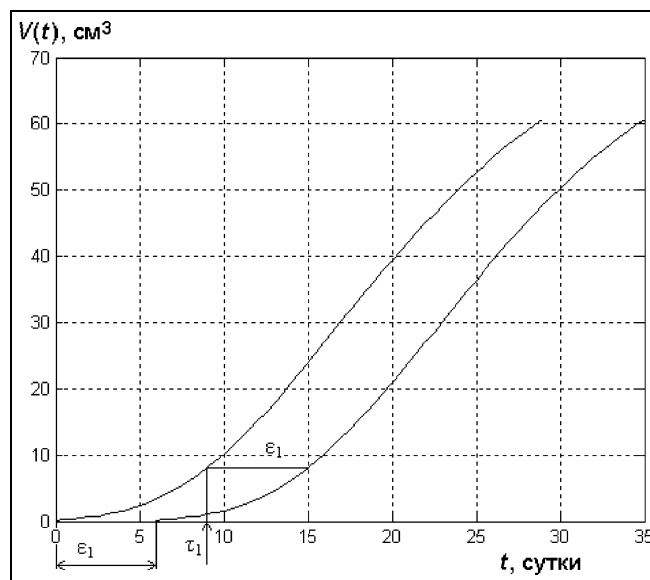


Рис. 2. Теоретические кривые роста опухоли в контроле и после введения препарата:

$\tau_1$  — момент первого введения препарата;  $\varepsilon_1$  — задержка роста опухоли;  $V$  — объем опухоли





уменьшение размеров опухоли происходит позднее момента введения и зависит от дозы препарата. Согласно принятой методике [18], при изучении противоопухолевой эффективности препарата степень уменьшения размеров опухоли оценивается по отношению ее размеров после введения препарата в опытной группе ( $P_{оп}$ ) к ее размерам в контрольной группе ( $P_k$ ). Отношение  $P_{оп}/P_k$ , % является количественной оценкой уменьшения размеров опухоли в некоторый фиксированный момент времени после терапевтического воздействия. Учитывая, что рост опухоли после проведенной терапии возобновляется со сдвигом по времени относительно контрольной кривой, длительность которого отражает эффективность лечения, значение длительности задержки роста опухоли также можно использовать для оценки эффективности химиотерапевтического воздействия [16, 17].

Приняв допущение о мгновенности гибели опухолевых клеток при введении препарата, можно описать рост опухоли после воздействия уравнением Гомпертца со сдвигом во времени на величину  $\varepsilon$ :

$$X(t) = X_0 \cdot \exp(\alpha[1 - \exp(-\beta(t - \varepsilon))]), \quad (3)$$

где  $\varepsilon$  — длительность задержки роста опухоли, которая оценивалась по экспериментальным кривым ее роста по завершении переходного периода после введения препарата (рис. 2). Этот параметр удобен для моделирования и позволяет вычислять число погибших клеток при введении препарата в зависимости от дозы  $D$  [16, 17].

Кроме параметра  $\varepsilon(D)$ , эффективность химиотерапевтического воздействия оценивалась на основании анализа параметров  $\alpha$  и  $\beta$ , значения которых отражают скорость процесса роста опухоли и ее конечный размер.

### Значения параметров функции Гомпертца для роста опухоли в контроле и при различных режимах введения доксорубина

Параметры	Контроль	Доза $D, M$				
		$1,4 \cdot 10^{-3}$	$10^{-5}$	$10^{-10}$	$10^{-15}$	$10^{-20}$
<i>Однократное введение</i>						
Моменты введения, сут	0	1				
Среднее время задержки, сут	—	5,26	6,62	5,36	5,61	7,03
Параметры Гомпертца:						
$\alpha$	6,08	6,16	6,24	8,71	7,99	9,52
$\beta$ , сут <sup>-1</sup>	0,06	0,06	0,09	0,10	0,16	0,12
$X_\infty$ , г	20,42	17,87	8,24	11,38	3,84	9,52
С. к. о.	0,0102	0,0057	0,0034	0,0384	0,0000	0,0290
<i>Трехкратное введение</i>						
Моменты введения, сут	0	1; 8; 15				
Среднее время задержки, сут	—	1,17	-0,17	0,11	-0,22	-0,10
Параметры Гомпертца:						
$\alpha$	6,08	3,37	5,45	5,65	5,90	5,38
$\beta$ , сут <sup>-1</sup>	0,06	0,10	0,09	0,10	0,10	0,10
$X_\infty$ , г	20,42	2,20	6,82	6,21	7,45	6,21
С. к. о.	0,0102	0,0004	0,0193	0,0076	0,0508	0,0140
<i>Четырехкратное введение</i>						
Моменты введения, сут	0	1; 5; 9; 13				
Среднее время задержки, сут	—	3,44	-1,39	0,48	0,11	2,29
Параметры Гомпертца:						
$\alpha$	6,08	3,76	5,49	5,41	5,77	4,36
$\beta$ , сут <sup>-1</sup>	0,06	0,08	0,10	0,10	0,10	0,09
$X_\infty$ , г	20,42	3,34	7,62	6,02	6,35	4,13
С. к. о.	0,0102	0,0131	0,0174	0,0093	0,0087	0,0047
<i>Пятикратное введение</i>						
Моменты введения, сут	0	1; 2; 3; 4; 5				
Среднее время задержки, сут	—	3,07	1,60	1,22	1,29	2,41
<i>После 5-го введения</i>						
Параметры Гомпертца:						
$\alpha$	6,08	4,64	4,34	4,69	4,37	4,22
$\beta$ , сут <sup>-1</sup>	0,06	0,13	0,11	0,17	0,14	0,12
$X_\infty$ , г	20,42	2,38	2,87	1,67	2,07	2,40
С. к. о.	0,0102	0,0018	0,0010	0,0028	0,0015	0,0012

Параметр  $\alpha$  связывает начальный размер опухоли  $X_0$  с конечным ее размером  $X_\infty$ , к которому стремится величина  $X(t)$  (см. уравнение (2)) при  $t \rightarrow \infty$ ,  $X_\infty = X_0 \cdot \exp \alpha$ . Учитывая, что конечная цель химиотерапевтического воздействия на опухоль состоит в уменьшении ее размеров в перспективе на удаленные сроки, параметр  $X_\infty$  также рассматривался в качестве показателя эффективности проведенной терапии.

## 2. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

Экспериментальные исследования по оценке эффективности различных режимов применения доксорубицина в стандартных терапевтических дозах и в сверхмалых дозах проводились в Институте биохимической физики и были доложены на III Международном симпозиуме “Механизмы действия сверхмалых доз” в Москве, 3–4 декабря 2002 г., а также опубликованы в различных журналах [12–14].

Материалы и методы исследования. Исследование проведено на 400 мышках, самках BDF1, массой 18–20 г (питомник “Столбовая”). Экспериментальной опухолью моделью служила карцинома легких Льюис, перевиваемая в соответствии со стандартной методикой измельченными фрагментами опухолевой ткани под кожу бока животного.

Доксорубицин вводился в диапазоне доз сверхнизких концентраций —  $10^{-5}$ ,  $10^{-10}$ ,  $10^{-15}$  и  $10^{-20}$  М, а также в стандартных терапевтических дозах, составляющих 2 мг/кг ( $0,35 \cdot 10^{-3}$  М) в случае пятикратного ежедневного применения препарата и 8 мг/кг ( $1,4 \cdot 10^{-3}$  М) при всех остальных изучавшихся режимах применения цитостатика.

Исследовано четыре схемы применения препарата:

- однократно;
- трехкратно с интервалом 6 суток между введениями препаратов;
- четырехкратно с интервалом в 4 суток между введениями препаратов;
- пятикратно, ежедневно.

При всех изученных схемах применения доксорубицин вводился внутривенно, начиная со следующих суток после перевивки опухоли, в виде водного раствора объемом 0,2 мл. Растворы препарата готовились методом последовательного многократного разведения с перемешиванием [12–14].

Исследовалась кинетика изменения размеров опухоли у контрольных и леченных животных. С этой целью на протяжении развития опухолевого процесса измерялись два взаимно перпендикулярных размера опухолевого узла (длина  $a$  и ширина  $b$ ). Объем опухоли вычислялся по размерам опухолевого узла как  $0,5ab^2$ , в соответствии с принятыми методическими рекомендациями по изучению активности противоопухолевых препаратов [18]. Учитывая, что плотность опухоли равен  $1 \text{ г/см}^3$ , вычисляемый объем опухоли принимался равным массе опухоли в граммах. Экспериментальная точка на кинетической кривой роста опухоли вычислялась как среднее арифметическое по группе из 6–8 животных. Раз-

брос значений массы опухоли в группе составлял не более 10 % от среднего.

### 2.1. Однократное введение

Анализ параметров функции Гомпертца показал, что при введении стандартной терапевтической дозы параметры  $\alpha$ ,  $\beta$  и  $X_\infty$  не изменяют своих значений по сравнению с контролем (см. таблицу). Это согласуется с результатами исследований кинетических кривых роста различных солидных опухолей после введения цитостатических препаратов при “высокодозовой” терапии [16]. В отличие от терапевтической дозы, однократное введение всех рассмотренных СМД приводит к возрастанию параметров  $\alpha$  и  $\beta$  и снижению конечного размера  $X_\infty$  опухоли по сравнению с контролем примерно в 5 раз. Возрастание параметра  $\alpha$ , согласно выражению  $X_\infty = X_0 \cdot \exp \alpha$ , должно приводить к увеличению конечного размера опухоли  $X_\infty$  при неизменном ее начальном размере  $X_0$ . Наблюдаемое уменьшение значения  $X_\infty$ , по-видимому, можно объяснить снижением значения  $X_0$ , что требует дополнительного экспериментального подтверждения.

Длительность задержки роста опухоли  $\varepsilon(D)$  как для терапевтической, так и для всех рассматриваемых СМД изменяется в пределах от 5 до 7 сут, что свидетельствует о сопоставимости эффектов их применения. Построенная зависимость “доза–эффект” имеет бимодальный характер (рис. 3, а) при наибольшей эффективности дозы  $10^{-20}$  М.

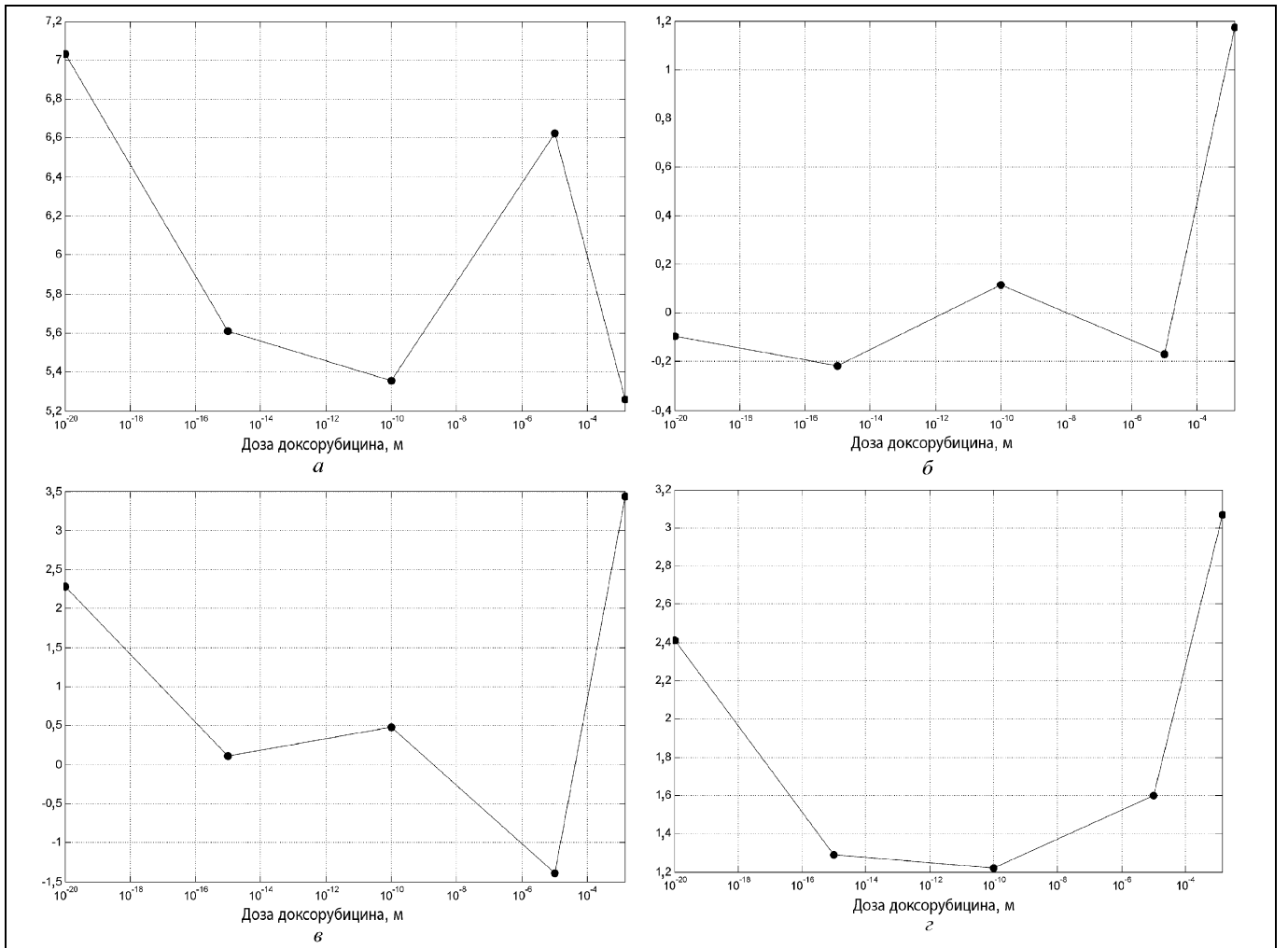
Кинетические кривые роста опухоли в контроле и после однократного введения доксорубицина в терапевтической дозе  $1,4 \cdot 10^{-3}$  М и в дозе  $10^{-20}$  М приведены на рис. 4.

### 2.2. Трехкратное введение

Параметры функции Гомпертца для трехкратного введения терапевтической дозы значительно отличаются от контрольных. Происходит снижение параметра  $\alpha$ , возрастание параметра  $\beta$  и уменьшение конечного размера опухоли  $X_\infty$  примерно в 9 раз по сравнению с контролем (см. таблицу). Снижение конечного размера опухоли  $X_\infty$  говорит о пролонгированном эффекте в отдаленные сроки роста опухоли.

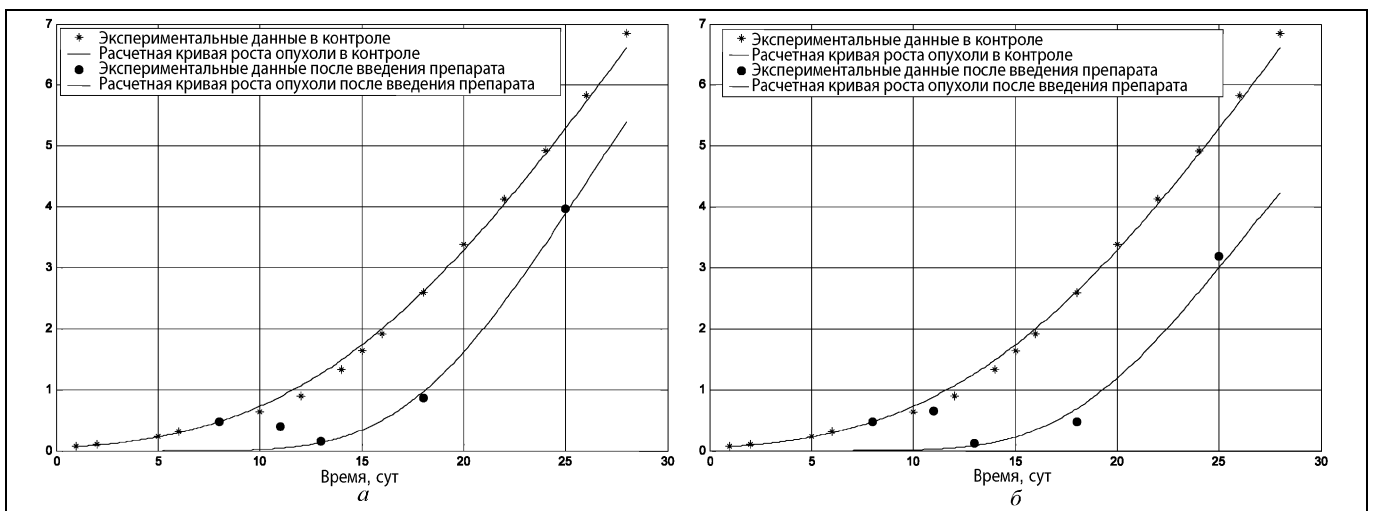
При введении СМД происходит снижение параметра  $\alpha$ , что существенно отличает данный режим от однократного введения препарата, при котором наблюдался рост значений этого параметра. Параметр  $\beta$  возрастает, повторяя тенденции режима однократного введения. Конечный размер опухоли  $X_\infty$  при введении СМД снижается примерно в 3 раза по сравнению с контролем, что значительно меньше, чем для терапевтической дозы (см. таблицу).

Анализ длительности задержки роста опухоли при трехкратном введении препарата показал, что эффективно только введение терапевтической дозы, так как приводит к положительному значению  $\varepsilon(D)$ , хотя значительно меньшему, чем при однократном ее введении. Применение СМД вызывает либо стимуляцию роста опухоли, о чем свидетельствуют отрицательные значения задержки роста  $\varepsilon(D)$ , либо эффект, близкий к нулевому. Следовательно, можно сделать вывод, что введе-



**Рис. 3. Зависимости “доза—эффект” при различных режимах введения доксорубина:**

однократное (а); трехкратное (б); четырехкратное (в) и пятикратное (г) поведения; по оси ординат — задержка роста опухоли в сутках



**Рис. 4. Кинетические кривые роста карциномы Льюис в контроле и после однократного введения доксорубина:**

а — в терапевтической дозе  $1,4 \cdot 10^{-3}$  М; б — в дозе  $10^{-20}$  М; по оси ординат — масса опухоли в граммах, по оси абсцисс — время после перевивки опухоли животным

ние доксорубина трикратно с интервалом в 6 суток не эффективно.

Построенная зависимость “доза—эффект” имеет так называемую “мертвую зону” для всего диапазона СМД и только терапевтическую дозу можно считать слабо эффективной (рис. 3, б). Кинетические кривые роста опухоли в контроле и после трехкратного введения доксорубина в терапевтической дозе  $1,4 \cdot 10^{-3}$  М и в дозе  $10^{-20}$  М приведены на рис. 5.

### 2.3. Четырехкратное введение

Анализ параметров функции Гомпертца для четырехкратного введения терапевтической дозы показывает снижение параметра  $\alpha$ , возрастание параметра  $\beta$  и уменьшение конечного размера опухоли  $X_{\infty}$  примерно в 7 раз по сравнению с контролем, что повторяет и усиливает тенденции режима трехкратного введения. Это

также свидетельствует о пролонгированном действии препарата в отдаленные сроки роста опухоли.

При введении СМД значение параметра  $\alpha$  снижается незначительно по сравнению с контролем. Параметр  $\beta$  возрастает, повторяя тенденции режима однократного и трехкратного введений. Конечный размер опухоли  $X_{\infty}$  также снижается, причем максимальное снижение в 5 раз по сравнению с контролем соответствует наиболее эффективной дозе  $10^{-20}$  М.

Анализ длительности задержки роста опухоли показал, что эффективность данного режима применения доксорубина выше, чем при трехкратном введении, но ниже, чем при однократном. Применение препарата в СМД эффективно только для дозы  $10^{-20}$  М, причем эффективность сравнима с действием терапевтической дозы (см. таблицу). Для дозы  $10^{-5}$  М длительность задержки отрицательна, что свидетельствует о стимуляции

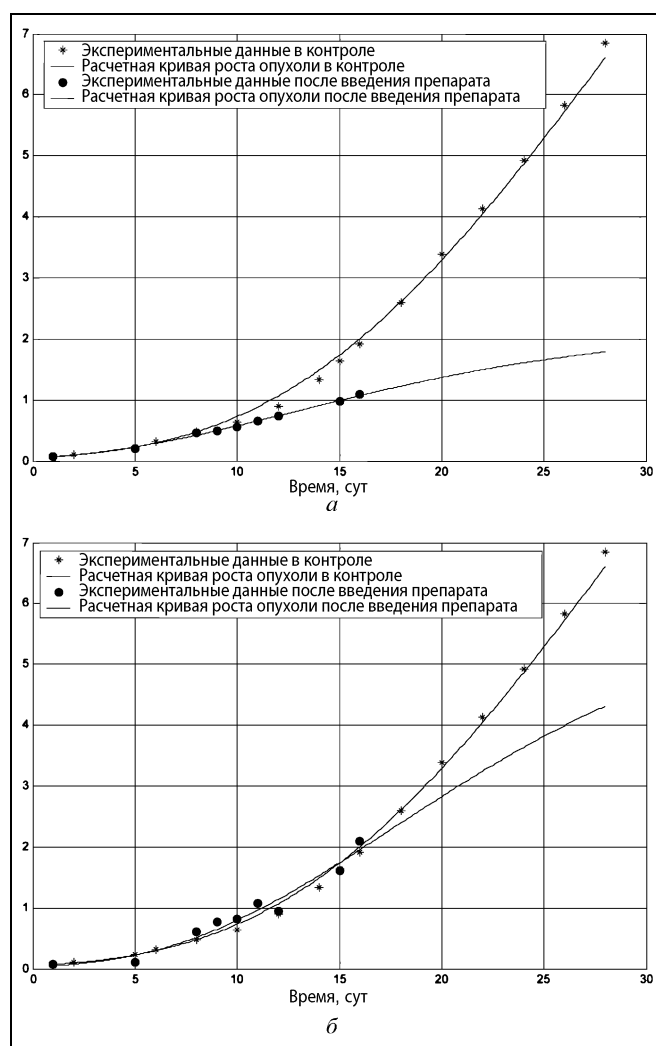


Рис. 5. Кинетические кривые роста карциномы Льюис в контроле и после трехкратного введения доксорубина:

$a$  — в терапевтической дозе  $1,4 \cdot 10^{-3}$  М;  $b$  — в дозе  $10^{-20}$  М; по оси ординат — масса опухоли в граммах; по оси абсцисс — время после перевивки опухоли животным

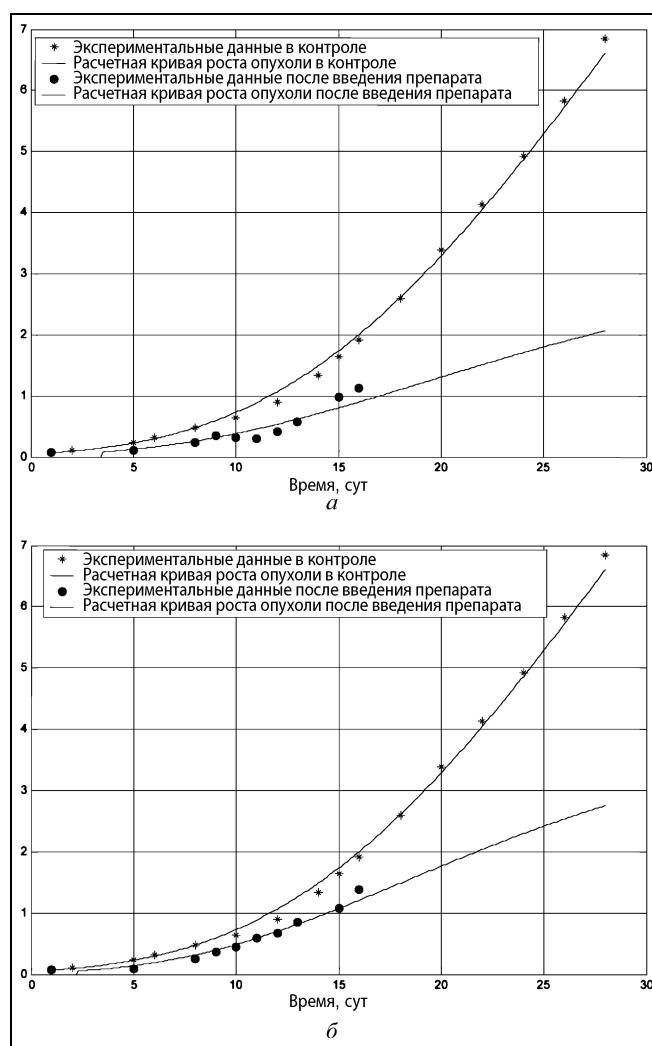


Рис. 6. Кинетические кривые роста карциномы Льюис в контроле и после четырехкратного введения доксорубина:

$a$  — в терапевтической дозе  $1,4 \cdot 10^{-3}$  М;  $b$  — в дозе  $10^{-20}$  М; по оси ординат — масса опухоли в граммах; по оси абсцисс — время после перевивки опухоли животным





роста опухоли, а дозы  $10^{-10}$  и  $10^{-15}$  М практически не эффективны. Анализ результатов моделирования позволяет сделать вывод о том, что четырехкратное введение доксорубина с интервалом в 4 сут более эффективно, чем трехкратное введение с интервалом в 6 сут.

Зависимость “доза—эффект” имеет бимодальный характер и свидетельствует о нарастании эффекта по мере уменьшения доз. Максимально эффективна в данном режиме доза  $10^{-20}$  М, причем ее эффективность сравнима с применением терапевтической дозы (рис. 3, в). Кинетические кривые роста опухоли в контроле и после четырехкратного введения доксорубина в терапевтической дозе  $1,4 \cdot 10^{-3}$  М и в дозе  $10^{-20}$  М приведены на рис. 6.

#### 2.4. Пятикратное введение

Анализ параметров функции Гомпертца, вычисленных по кинетическим кривым роста опухоли в контроле и после пятикратного ежедневного введения препарата как в терапевтических, так и в сверхмалых дозах подтверждает отмеченные ранее тенденции уменьшения конечных размеров опухоли  $X_{\infty}$  примерно в 10 раз по сравнению с контролем, увеличение параметра  $\beta$  в 2—3 раза и снижение значений параметра  $\alpha$  примерно в 1,5 раза по сравнению с контролем (см. таблицу). Изменения всех параметров функции Гомпертца значительно больше, чем при трехкратном и четырехкратном режимах. Это свидетельствует о значительном изменении характера роста опухоли. Этот эффект имеет место как для терапевтической, так и для сверхмалых доз.

Зависимость “доза—эффект” имеет “бимодальный” характер с нарастанием эффекта при уменьшении дозы, что имело место, как для четырехкратного, так и для однократного введения доксорубина (рис. 3, г). Максимально эффективна в данном режиме доза  $10^{-20}$  М, причем ее эффективность сопоставима с действием терапевтической дозы.

#### ЗАКЛЮЧЕНИЕ

Применение математического моделирования для описания кинетики роста опухоли позволило оценивать эффективность химиотерапевтического воздействия по изменению значений параметров модели и по длительности задержки роста опухоли. Длительность задержки роста опухоли отражает эффекты, связанные с гибелью клеток непосредственно после воздействия и наиболее четко определяется при однократном введении препарата. Конечный размер опухоли, отражающий изменение вида кинетических кривых роста опухоли, характеризует прогноз течения опухолевого процесса на поздние сроки его развития и наиболее точно оценивает эффективность режимов многократного введения препарата.

Полученные оценки эффективности доксорубина с использованием параметров модели подтверждают результаты, полученные в Институте биохимической физики РАН [12—14], когда показателем противоопухолевого эффекта служил общепринятый параметр, характеризующий отношение средних размеров опухоли у леченых животных к контролю [12—14, 18].

При однократном введении доксорубина четко прослеживается различие в действии стандартной терапевтической дозы и сверхмалых доз. Введение терапевтической дозы не приводит к изменению параметров роста опухоли, вызывая только задержку ее роста, связанную с гибелью опухолевых клеток, что и отражает ее противоопухолевый эффект. Введение СМД вызывает не только задержку роста опухоли, но и изменение параметров ее роста, что свидетельствует о наличии дополнительных механизмов воздействия на опухолевый процесс, которые снижают ее конечный размер в отдаленные сроки.

При многократных режимах с использованием трех и четырехкратных введений доксорубина с интервалом в 6 и 4 сут, соответственно, эффективность СМД резко снижается по сравнению с терапевтической дозой. Причем некоторые СМД приводят к стимуляции роста опухоли. Объяснения причин такой реакции опухоли невозможны без понимания механизмов действия СМД.

При ежедневном пятикратном введении доксорубина, как для терапевтической дозы, так и для СМД, эффект был практически идентичен как по длительности задержки роста опухоли, так и по конечному размеру опухоли.

Оценка эффективности терапии по длительности задержки роста опухоли позволила выявить закономерность нарастания эффекта при уменьшении СМД в диапазоне от  $10^{-5}$  до  $10^{-20}$  М. Эта закономерность не прослеживается только для режима трехкратного введения, при котором применение СМД приводит к стимуляции роста опухоли. Этот результат требует дополнительного исследования. Построенные зависимости “доза—эффект” являются бимодальными и свидетельствуют о сопоставимости эффекта действия доксорубина в терапевтической дозе  $1,4 \cdot 10^{-3}$  М и в дозе  $10^{-20}$  М.

Накопленные экспериментальные данные об эффектах воздействия СМД различных препаратов и биологически активных соединений требуют разработки теории, объясняющей наблюдаемые результаты. До настоящего времени нет общепринятой точки зрения на механизмы и процессы, лежащие в основе действия СМД. Однако имеется ряд гипотез, которые отражают направления поиска этих механизмов в области биохимии, биофизики, физиологии и фармакологии.

В основе большинства гипотез возможные механизмы действия СМД рассматриваются с позиций молекулярной рецепции. Так, в основе одной из гипотез лежат представления о наличии высокоэффективных систем усиления сигнала, который формируется благодаря протеканию каскадных процессов. Предполагается, что для достижения ответа на действие эффектора достаточно, чтобы он связался только с незначительной частью рецепторов, имеющихся на клетке [19, 20].

Гипотеза, разрабатываемая в работах [7, 11], предполагает формирование ответа в условиях неравновесного связывания лиганда с рецептором. Предполагается, что для достижения биоэффекта достаточно того, чтобы самые “быстрые” молекулы действующего вещества достигли клеточной мишени.

В работах [21—23] предложена интерпретация эффектов СМД на основе адаптационного механизма, поскольку адаптация приводит к тому, что клетка может

реагировать не на саму концентрацию вещества, а на изменение его концентрации.

Все перечисленные подходы можно применять только для доз, больших  $10^{-23}$  М. Известно, что число Авогадро примерно равно  $6 \cdot 10^{23}$  моль $^{-1}$ . Значит для доз менее  $10^{-24}$  М во всем экспериментальном объеме жидкости может не быть ни одной молекулы действующего вещества. Следовательно, для доз менее  $10^{-23}$  М наблюдаемые эффекты с позиций молекулярной рецепции не могут быть объяснены.

С учетом этого обстоятельства в ряде работ рассматривается гипотеза, объясняющая эффект действия биологически активных веществ в сверхнизких дозах с позиций анализа молекулярной структуры воды и ее водородных связей [24, 25]. Анализ различных состояний воды выявил наличие информационного характера химических взаимодействий и возможность передачи информации в живых системах путем изменения структурных состояний водной среды внутри организма. Результаты исследований показали, что можно изменять ориентацию части структурных элементов воды в зависимости от характера воздействия. Новое расположение структурных элементов воды служит формой записи информации о произведенном воздействии, и вода может выступать в качестве молекулярного информационного ретранслятора с соседними элементами.

Кроме приведенных гипотез, существует множество и других, но ни одна из них пока не является строгой теорией, объясняющей механизм действия веществ в сверхмалых дозах.

## ЛИТЕРАТУРА

1. Подколзин А. А., Гуревич К. Г. Действие биологически активных веществ в малых дозах. — Изд-во КМК, 2002. — 170 с.
2. Действие малых доз нитрозометилмочевины на стационарную клеточную популяцию. Эксперимент и моделирование / Т. В. Крутова, С. Б. Черникова, А. А. Конрадов, Е. Б. Бурлакова // Известия РАН, сер. Биол. — 1992. — № 4. — С. 511.
3. Bonavida B. Citotoxicite antitumoral a tris faible dose // In: Abstr. of the Second Intern. Congr. on Ultra Low Doses / France, Bordeaux, 1993. — P. 12.
4. Burlakova E. B. Some specific aspect of ultra low dose effect of chemical compaunds and physical agents // In: Abstr. of the Second Intern. Congress on Ultra Low Doses / Ibid. — P. 16.
5. Doutremepuiech C. Etude de l'aspirine a tres faible dose // In: Abstr. of the Second Intern. Congress on Ultra Low Doses / Ibid. — P. 2.
6. Effect of ethylnitrosourea on the stationary cell population / T. V. Krutova, S. B. Chernikova, A. A. Konradov, E. B. Burlakova // In: Abstracts. of the Second Intern. Congress on Ultra Low Doses / Ibid. — P. 22.
7. Бурлакова Е. Б., Конрадов А. А., Худяков И. В. Воздействие химических агентов в сверхмалых дозах на биологические объекты // Изв. АН СССР, сер. Биол. — 1990. — № 2. — С. 184—193.
8. Benveniste J., Davenas E. L'agiacion de solutions hantement diluees n'induit pas d'activite biologique specifique // C.R. Acad. Sci. Paris., 1991. — P. 461—466.
9. The simila principle / P. Bellavite, S. Lussignolli, M.L. Semizzi, R. Ortolani, A. Signorini // British Homoeopathic Journal. — 1997. — Vol. 86. — P. 73—85.
10. Блюменфельд Л. А. Понятие конструкции в биологической физике. К вопросу о механизме действия сверхмалых доз // Российской химический журнал. — 1999. — Т. 43, № 5. — С. 15—20.
11. Бурлакова Е. Б. Особенность действия сверхмалых доз биологически активных веществ и физических факторов низкой интенсивности // Там же. — С. 3—11.
12. Сверхмалые дозы доксорубина: ингибирование опухолевого роста в эксперименте / Л. А. Островская, Н. В. Блюхтерова, М. М. Фомина и др. // Бюллетень экспериментальной биологии и медицины. — 2002. — Приложение № 4. — С. 52—54.
13. Возможности использования сверхмалых доз цитостатиков в химиотерапии опухолей / Л. А. Островская, Н. В. Блюхтерова, М. М. Фомина и др. // Наука — производству. — 2002. — № 3. — С. 59—60.
14. Чувствительность экспериментальных опухолевых моделей к сверхмалым дозам доксорубина / Островская Л. А., Блюхтерова Н. В., Фомина М. М. и др. // Радиационная биология. Радиология. — 2003. — Т. 43, № 3. — С. 273—281.
15. Эмануэль Н. М. Кинетика экспериментальных опухолевых процессов. — М.: Наука, 1977. — 419с.
16. Бабушкина Н. А. Математическое моделирование сложных режимов химиотерапии экспериментальных опухолей: Дис. канд. биол. наук. — М.: ИБХФ, 1985. — 189 с.
17. Иванов В. К., Петровский А. М. Количественная оценка выживаемости клеточных популяций при фракционированном облучении // Мед. радиология. — 1982. — Т. 27, № 3. — С. 77—80.
18. Методические рекомендации по предклиническому изучению специфической активности противоопухолевых препаратов, предлагаемых для испытания в клинике. — М.: Минздрав РФ, 2000.
19. Гуревич К. Г. Вероятностное описание системы "лиганд—рецептор" // Биофизика. — 1999. — Т. 44, вып. 6. — С. 1022—1026.
20. Варфоломеев С. Д., Гуревич К. Г. Биокинетика. Практический курс. — М.: Фаир-пресс, 1999. — 716 с.
21. Сазанов Л. А., Зайцев С. В. Действие сверхмалых доз ( $10^{-18}$ — $10^{-14}$ ) биологически активных веществ: общие закономерности, особенности и возможные механизмы // Биохимия. — 1992. — Т. 57, вып. 10. — С. 1443—1460.
22. Зайцев С. В., Ефанов А. М., Сазанов Л. А. Общие закономерности и возможные механизмы действия биологически активных веществ в сверхмалых дозах // Российской химический журнал. — 1999. — Т. 43, № 5. — С. 28—33.
23. Zaitsev S. V., Il'ina A. D., Varfolomeev S. D. // In: Research in Biochemical Kinetics. — N.-Y.: Nova Science Publishers, Inc., 1993. — P. 102—119.
24. Зенин С. В., Тяглов Б. В. Природа гидрофобного взаимодействия. Возникновение ориентационных полей в водных растворах // Журн. физ. хим. — 1994. — Т. 68, № 3. — 500 с.
25. Зенин С. В., Полануер Б. М., Тяглов Б. В. Экспериментальное доказательство наличия фракций воды // Гомеопатическая медицина и акупунктура. — 1998. — № 2. — С. 41.

# СТРУКТУРНО-ТЕХНОЛОГИЧЕСКИЙ РЕЗЕРВ И ЕГО ИСПОЛЬЗОВАНИЕ ДЛЯ ПОВЫШЕНИЯ УСТОЙЧИВОСТИ ПРОИЗВОДСТВЕННЫХ СИСТЕМ

М. Ю. Копнин<sup>(1)</sup>, В. В. Кульба<sup>(1)</sup>, Е. А. Микрин<sup>(2)</sup>

<sup>(1)</sup> Институт проблем управления им В. А. Трапезникова, г. Москва;

<sup>(2)</sup> Ракетно-космическая корпорация "Энергия", г. Королёв

Введено понятие одного из специальных видов резервов, характерных для производственных систем, — структурно-технологического резерва. Для описания работы таких систем применен язык модифицированных сетей Петри. На основании свойств потока маркеров введены понятия стоимости, гибкости и устойчивости производственного процесса.

## ВВЕДЕНИЕ

Возрастание масштабов техногенной деятельности современного общества крайне обострили проблемы, связанные с обеспечением безопасности человека, общества и окружающей среды. В этой связи крайне актуальной становится задача повышения устойчивости производственных систем к возникающим на них авариям [1]. В данной статье приводится описание технологии работы производственной системы на языке модифицированных сетей Петри [2—4], представляющее собой последовательность срабатывания переходов, задающую некоторый поток маркеров.

В ходе производственного процесса на отдельных звеньях могут возникать различного рода аварии, несчастные ситуации, приводящие к возникновению поражающих факторов, и прочие нежелательные события, которые могут быть классифицированы как чрезвычайная ситуация (ЧС). В результате функционирования части производственных звеньев может быть остановлено [5, 6]. Для возобновления производственного процесса в целом, возможно, с меньшей эффективностью, необходимо наличие резерва. В данной статье определяется один из специальных видов резервов — структурно-технологический резерв, вводятся понятия стоимости, гибкости и устойчивости производственного процесса.

## 1. ОПИСАНИЕ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ

Вначале дадим нестрогие определения рассматриваемых понятий, в дальнейшем те же определения будут сформулированы на языке сетей Петри.

*Технологической операцией*, выполняемой на производственном звене, назовем преобразование сырья в го-

товую продукцию. Результат одной технологической операции может служить входом для другой. Таким образом, *технологический процесс* может быть задан как набор взаимосвязанных по входам и выходам технологических операций.

У каждой технологической операции может быть несколько *технологических схем* — способов преобразования сырья в готовую продукцию. Кроме того, для каждой технологической схемы может быть задан вектор характеристик, таких как время выполнения технологической операции, стоимость операции и др.

*Структурно-технологическим резервом технологической схемы операции* назовем множество технологических операций и их технологических схем (оборудования и технологии его применения), которые могут заменить данную технологическую схему операции при выходе ее строя, не нарушая работу системы (технологического процесса) в целом.

*Структурно-технологическим резервом системы* назовем множество технологических операций и их технологических схем, без которых система в целом может продолжать функционирование, возможно, с меньшей эффективностью. Структурно-технологический резерв системы — это множество ресурсов различных видов, за счет которых достигается технологическая гибкость системы, т. е. способность системы компенсировать технологические возмущения, такие как отказы и повреждение технологических звеньев, разрыв транспортных связей и т. д.

Теперь дадим определения рассматриваемых понятий в терминах модифицированных сетей Петри.

Рассмотрим нециклический технологический процесс, который задан как набор взаимосвязанных технологических операций  $S_i$ . Виды используемого в данном

технологическом процессе сырья и получаемой продукции обозначим как  $\omega_j, j = \overline{1, J}$ .

Построим сеть Петри  $N = \langle P, T, F \rangle$ , реализующую данный технологический процесс. Каждому производственному звену  $s_i, i = \overline{1, S}$  поставим в соответствие переход  $t_i \in T$ , позициям  $p_j^i \in P$  соответствуют некоторые емкости, в которых может находиться сырье или готовая продукция вида  $\omega_j$ . Функциональное отличие переходов от позиций состоит в том, что в позициях, в отличие от переходов, не происходит преобразований изделий одного вида в изделия другого вида.

В нециклическом процессе можно выделить входную  $p_{\text{вх}}$  и выходную позиции  $p_{\text{вых}}$ , на первую из них поступает сырье, а на вторую — готовая продукция — результат технологического процесса.

**Определение 1.** Технологической операцией  $s_i$ , выполняемой на производственном звене, назовем функции входного и выходного распределений маркеров на переходе  $t_i$ .

Множество всех технологических операций обозначим через  $S$ . Для каждой операции могут существовать несколько способов преобразования входного распределения маркеров в выходное распределение, задающихся парой функций  $\lambda$  и  $\varphi$ . Для каждого способа преобразования может быть также задан вектор характеристик  $\bar{\tau}$ , таких как время выполнения операции, стоимость выполнения и др.

Для одной и той же операции  $s_i$  может быть несколько возможных технологических схем  $T_i^k$ .

**Определение 2.** Технологической схемой  $T_i^k$  операции  $s_i$  назовем тройку функций:

$$T_i^k = \begin{cases} \lambda_i^k(p) - \text{функция входного распределения маркеров,} \\ \varphi_i^k(p) - \text{функция выходного распределения маркеров,} \\ \bar{\tau}_i^k - \text{вектор характеристик технологической схемы.} \end{cases}$$

В силу ацикличности технологического процесса, для каждой технологической операции  $s_i$  могут быть выделены два непересекающихся множества  $P_i' = \{p_r'\}$ ,  $r = \overline{1, n}$ , и  $P_i'' = \{p_s''\}$ ,  $s = \overline{1, m}$ , входных и выходных позиций. В сети Петри технологическая схема  $T_i^k$  для операции  $s_i$  задается переходом  $t_i^k$  следующим образом (рис. 1):

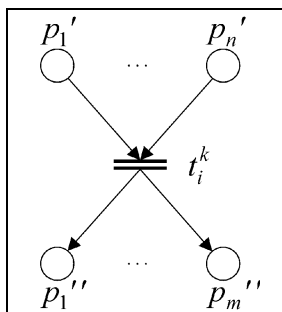


Рис. 1. Представление технологической схемы в сети Петри

При этом функция  $\lambda_i^k$  определяет кратность входных дуг для перехода  $t_i^k$ , а функция  $\varphi_i^k$  — кратность выходных.

Содержательно, функция  $\lambda_i^k$  задает минимальное количество ресурсов, необходимое для выполнения технологической операции, а функция  $\varphi_i^k$  — количество ресурсов, получаемое из этого минимального количества.

Вектор  $\bar{\tau}_i^k$  также определяет ряд характеристик сети, такие как время выполнения операции, стоимость выполнения операции и др.

**Определение 3.** Технологической схемой процесса назовем такой набор технологических схем операций, при которых достижима конечная маркировка.

Конечной маркировкой в сети является маркировка, в которой в завершающей позиции  $p_{\text{вых}}$  присутствуют маркеры.

Технологической схемой процесса в смысле сетей Петри служит некоторое слово  $l_i$  языка сетей Петри. Множество всех слов обозначим через  $L = \{l_i\}, i = 1$ .

## 2. СТОИМОСТЬ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА

Будем рассматривать маркеры, присутствующие во входном и выходном распределениях технологической схемы  $T_i^k$  операции  $s_i$  как ресурсы, имеющие стоимостные характеристики  $C(p_j) \geq 0$ . Пусть элементы вектора характеристики операции  $\tau_i^{k(l)} \in \bar{\tau}_i^k$  также имеют стоимость  $C(\tau_i^{k(l)}) \geq 0$ . При выполнении операции естественно стремиться к увеличению стоимости ресурсов на выходе при уменьшении стоимости ее выполнения и стоимости ресурсов на входе.

Обслуживание даже не задействованного в данный момент оборудования и поддержание его в работоспособном состоянии требует материальных затрат. Обозначим эти затраты через  $\tau_i^{k(\text{обс})} \in \bar{\tau}_i^k$ .

**Определение 4.** Стоимостью операции  $s_i$  при использовании технологической схемы  $T_i^k = \{\lambda_i^k(p), \varphi_i^k(p), \bar{\tau}_i^k\}$  будем называть величину  $C(s_i, T_i^k) = \sum_{p_j \in P} C(p_j) \cdot \varphi_i^k(p_j) -$

$$- \sum_{p_m \in P} C(p_m) \cdot \lambda_i^k(p_m) - \sum_{\tau_i^{k(l)} \in \bar{\tau}_i^k} C(\tau_i^{k(l)}) - \sum_{q=1, q \neq k}^{n_i} C(\tau_i^{q(\text{обс})}),$$

где  $n_i$  — число технологических схем операции  $s_i$ .

Таким образом, общая стоимость технологической схемы операции складывается из стоимости ресурсов на выходе минус стоимость входных ресурсов, минус стоимость выполнения операции и минус стоимость обслуживания всех ее технологических схем.

Стоимость незадействованной в данный момент операции  $C(s_i, T_i^0) = - \sum_{q=1}^{n_i} C(\tau_i^{q(\text{обс})})$ .

**Определение 5.** Суммарной функцией стоимости технологического процесса назовем сумму стоимостей всех операций, выполненных в процессе достижения конечного





результата  $C(l) = \sum_{t_i^k \in l} C(s_i, T_i^k) - \sum_{t_j^q \in \bar{l}} C(\tau_i^{q(\text{обс})})$ , где  $l$  — слово языка сети Петри  $N$ , полученное при достижении конечной маркировки, а  $\bar{l}$  — множество переходов из операций, которые не были задействованы при достижении конечной маркировки.

Слов  $l$  может быть несколько, и естественно использовать в технологическом процессе такую последовательность срабатывания переходов-операций  $l_0$ , при которой достигается максимум:  $C_0 = \max_{l \in L} C(l)$ .

В дальнейшем будем предполагать, что используется именно эта схема, и возникающие аварийные ситуации затрагивают работу одного из звеньев, задействованных в данной схеме.

При нормальном функционировании технологической системы суммарная функция стоимости по всем ее операциям в большинстве случаев должна быть величиной неотрицательной из соображений рентабельности. При достижении этой функцией некоторого отрицательного порога можно говорить, что в системе сложилась ЧС. Кроме того, ЧС может возникнуть на отдельных операциях технологического процесса. Стоимость операций на данных участках технологической схемы может стать отрицательной, при этом стоимость для процесса в целом может оставаться положительной. Положительная стоимость процесса также может быть достигнута при использовании резервной технологической схемы. В таком случае будем говорить, что система обладает достаточным резервом при сложившейся ЧС, т. е. несмотря на ЧС на отдельных операциях, система продолжает функционировать без убытков. Кроме того, максимизируя стоимость процесса  $C(l)$  по всем резервным технологическим схемам, можно выбрать оптимальную резервную технологическую схему при имеющихся повреждениях.

При возникновении ЧС в масштабах всей технологической системы, ее стоимость  $C(l)$  принимает отрицательное значение либо за счет прекращения выдачи изделий на выходе, либо из-за увеличения времени выполнения операций. Если конечная маркировка становится недостижимой, то можно считать, что  $C(l) = - \sum_{t_j^k \in T} C(\tau_i^{q(\text{обс})})$  — стоимость невыполненного процесса.

Поскольку при невозможности выполнения той или иной технологической схемы может потребоваться перестроение части технологической схемы процесса, то представляется целесообразным ввести понятие стоимости смены технологической схемы, которая используется для выполнения операции.

Через  $\rho(s_i, T_i^1, T_i^2)$  будем обозначать затраты, требуемые для перехода с технологической схемы  $T_i^1$  на схему  $T_i^2$  для операции  $s_i$ . Эта величина будет определяться затратами на монтажные, пусконаладочные и прочие виды работ, связанных со сменой технологической цепочки. Для удобства расчетов доопределим эту функцию, положив  $\rho(s_i, T_i^1, T_i^2) = 0$ .

**Определение 6.** Функцией стоимости смены схемы технологической схемы  $T_i^1$  на схему  $T_i^2$  для операции  $s_i$  назовем величину  $C(s_i, T_i^1, T_i^2) = C(s_i, T_i^1) + \rho(s_i, T_i^1, T_i^2) - C(s_i, T_i^2)$ .

Таким образом, общая стоимость смены схемы складывается из стоимости исходной технологической схемы операции, плюс затраты на перенастройку и минус стоимость конечной технологической схемы операции.

Стоимость смены схемы будет тем больше, чем больше разность между стоимостью исходной и конечной технологическими схемами, а также чем больше затраты на перенастройку. При выборе схемы, на которую осуществляется переход, следует минимизировать рассматриваемую стоимость.

На основании понятия стоимости смены технологической схемы операции введем функцию стоимости смены схемы технологического процесса.

**Определение 7.** Функцией стоимости смены схемы технологического процесса назовем сумму стоимостей смены схем технологических операций процесса:  $C(l', l'') =$

$$= \sum_{t_i^k \in l'} C(s_i, T_i^k) + \sum_{i=1}^S \rho_i(l', l'') - \sum_{t_i^k \in l''} C(s_i, T_i^k), \text{ где слово } l' \text{}$$

задает исходную схему технологического процесса, слово  $l''$  — схему, на которую осуществляется переход, а  $\rho_i(l', l'')$  — стоимость смены технологической схемы для операции  $s_i$ .

Эта стоимость рассчитывается по правилу:

$$\rho_i(l', l'') = \begin{cases} \rho(s_i, T_i^{j_1}, T_i^{j_2}), & \text{если } \exists j_1, j_2, 1 \leq j_1, \\ & j_2 \leq n_i : t_i^{j_1} \in l' \text{ и } t_i^{j_2} \in l'' \\ \rho(s_i, T_i^{j_1}, T_i^0), & \text{если } \exists j_1, 1 \leq j_1 \leq n_i : t_i^{j_1} \in l', \\ & \text{и } \forall j_2, 1 \leq j_2 \leq n_i : t_i^{j_2} \notin l'' \\ \rho(s_i, T_i^0, T_i^{j_2}), & \text{если } \exists j_2, 1 \leq j_2 \leq n_i : t_i^{j_2} \in l'', \\ & \text{и } \forall j_1, 1 \leq j_1 \leq n_i : t_i^{j_1} \notin l'' \\ 0 & \text{— иначе.} \end{cases}$$

После выхода из строя звена  $s_j$ , входящего в используемую изначально схему процесса, и появления на этом звене поражающего фактора, потребуется переход на другую схему производства. В качестве заменяющей схемы может быть выбрана та допустимая схема, переход на которую потребует меньших затрат:  $\min_{l \in L, t_j^k \notin l} C(l_0, l)$ .

В этом выражении требование  $t_j^k \notin l$  означает невозможность использования технологических схем вышедшего из строя звена  $s_j$ .

Недостаток такого метода выбора резервной схемы технологического процесса состоит в отсутствии учета возможности повторного возникновения ЧС на других производственных звеньях.

### 3. СТРУКТУРНО-ТЕХНОЛОГИЧЕСКИЙ РЕЗЕРВ И ЕГО ХАРАКТЕРИСТИКИ

В случае возникновения аварии может сложиться ситуация, в которой станет невозможным полностью или частично выполнять одну или несколько операций. И, как следствие аварии, выполнение части других операций производственного процесса станет невозможным или нецелесообразным. Множество таких операций и их технологических схем будем называть *аварийным множеством*. Множество операций, дублирующих выполнение вышедших из строя звеньев, будем называть *структурно-технологическим резервом* или *резервным множеством*.

Пусть  $L = \{l\}$  — множество слов сети Петри. Каждому слову из него соответствует некая схема технологического процесса, а сами слова содержат последовательности сработавших переходов (примененных технологических схем) при достижении конечной маркировки.

Если применение используемой ранее на операции  $s_i$  технологической схемы  $T_i^n$  невозможно, то множество  $L$  разбивается на два непересекающихся подмножества:  $L = L_A \cup L_P$ , где  $L_A$  — множество слов, в которые входит переход  $t_i^n$ . Оно содержит те схемы процессов, которые станут неприменимыми после возникновения ЧС и при невозможности выполнения схемы  $T_i^n$  на операции  $s_i$ .  $L_P$  — множество слов, в которые переход  $t_i^n$  не входит. Оно содержит те схемы процессов, которые могут резервировать использование схемы  $T_i^n$  на операции  $s_i$ .

Как видно из построения, множества  $L_A$  и  $L_P$  не пересекаются.

Введем классификацию для переходов сети. Для этого рассмотрим их вхождения в слова из  $L_A$  и  $L_P$ . Возможны три класса переходов.

- Переходы, которые содержатся в словах из обоих множеств. Они могут выполняться как до возникновения ЧС, так и после. Исключим их из дальнейшего рассмотрения.
- Переходы, которые содержатся в словах из множества  $L_A$ , но не содержатся в словах из множества  $L_P$ . Эти переходы образуют *аварийное множество*  $R_A$ . Выполнение технологических схем, соответствующих этим переходам, становится невозможным или нецелесообразным.
- Переходы, которые содержатся в словах множества  $L_P$ , но не содержатся в словах из множества  $L_A$ . Они образуют *резервное множество*  $R_P$ . Его элементы выполняют дублирующую функцию в случае аварии.

Заметим, что в сети не существует переходов, не входящих ни в одно из множеств, поскольку такие переходы соответствовали бы неприменимым изначально технологическим схемам.

Аналогичным образом можно построить аварийное и резервное множества для операции  $s_i$  и совокупности технологических схем  $T_i^{i_1}, \dots, T_i^{i_n}$ , а также для операции в целом. Для этого при разделении множества слов  $L$  на два подмножества в качестве критерия разделения сле-

дует использовать наличие в словах  $l$  переходов  $t_i^{i_1}, \dots, t_i^{i_n}$  в первом случае или всех переходов  $t_i^{i_1}, \dots, t_i^{i_n}$  операции  $s_i$  во втором случае.

Введем некоторые характеристики сети, моделирующей производственную систему. Если после удаления из сети Петри  $N$  всех переходов, принадлежащих аварийному множеству  $R_A(s_i)$ , в полученной таким образом сети  $N_A$  остается достижимой конечная маркировка, то это означает, что рассматриваемая система обладает резервом при невозможности выполнения операции  $s_i$ . Набор последовательностей срабатывания переходов усеченной сети  $N_A$ , которые ведут к достижению конечной маркировки, назовем *резервной технологической схемой* производственной системы при невозможности выполнения операции  $s_i$  и будем обозначать как  $L_P(s_i)$ .

**Определение 8.** *Структурно-технологическим резервом системы (комплекса операций), заданной сетью Петри  $N$ , назовем множество переходов  $R_N = \{t_i^k\}$  такое, что при удалении любого из этих переходов из исходной сети  $N$  остается достижимой конечная маркировка сети  $N$ .*

При выходе из строя любого единичного элемента, принадлежащего множеству  $R_N$ , технологический процесс продолжает выполняться, возможно, с меньшей эффективностью. При выходе из строя любого числа элементов, не принадлежащих множеству  $R_N$ , система перестает функционировать. Если повреждено более одной операции из множества  $R_N$ , то дать однозначный ответ относительно работоспособности системы нельзя.

**Определение 9.** *Множество  $\bar{R}_N$  (дополнение к множеству  $R_N$ ) назовем множеством уязвимости системы (комплекса операций), заданной сетью  $N$ .*

В него войдут те элементы, повреждение которых влечет за собой выход из строя всей системы. Множество  $\bar{R}_N$  может быть получено включением в него тех и только тех переходов, которые принадлежат всем словам из языка  $L$ .

На множестве  $\bar{R}_N$  вводятся различные характеристики: число операций; число операций, отнесенное к общему числу операций системы; суммарная эффективность операций из этого множества при использовании любых технологических схем на этих операциях и др.

Введем ряд характеристик структурно-технологического резерва операции  $s_i$ .

**Определение 10.** *Множеством применимости структурно-технологического резерва операции  $s_i$  назовем множество переходов сети  $\Gamma(s_i) = \{t_j^n\}$  таких, что найдется технологическая схема  $T_i^k$  рассматриваемой операции  $s_i$  и соответствующий этой схеме переход  $t_i^k \in R_P(s_i, T_j^n)$ .*

Другими словами, это множество технологических схем операций, для замены которых могут использоваться технологические схемы операции  $s_i$ .

Как было показано ранее, возможны различные варианты реагирования производственной системы на удаление из нее одной из операций — от понижения эф-



фективности до полного останова. Введем количественные характеристики этого реагирования.

**Определение 11.** Гибкостью  $G(s_i)$  производственной системы относительно невозможности выполнения операции  $s_i$  назовем отношение числа технологических схем процесса  $|L_p(s_i)|$ , способных выполняться при невозможности выполнения операции  $s_i$ , к общему числу технологических схем процесса  $|L|$ , выполнимых при нормальном функционировании системы:  $G(s_i) = |L_p(s_i)|/|L|$ .

Как видно из определения, значение гибкости — отрицательное число, не превосходящее единицы. Если гибкость  $G(s_i) = 0$ , то при невозможности выполнения операции  $s_i$  вся система выходит из строя. Если гибкость  $G(s_i) = 1$ , то это значит, что операция  $s_i$  не используется ни в одной технологической схеме процесса и может быть без всякого ущерба удалена из производственной системы. Чем больше значение гибкости, тем меньше возникает потерь из-за невозможности выполнения операции  $s_i$ .

Для изучения гибкости системы в целом можно пользоваться значением средней гибкости:  $G_{cp} =$

$$= \sum_{i=1}^S P(A_i) G(s_i), \text{ где } P(A_i) \text{ — вероятность отказа звена } s_i.$$

По аналогии с гибкостью относительно невозможности выполнения одной операции, можно утверждать, что чем больше значение средней гибкости системы, тем меньше возникает потерь из-за невозможности выполнения какой-либо операции.

Недостаток данного определения гибкости состоит в том, что оно не учитывает стоимостных характеристик потерь в результате ЧС. При расчете гибкости производственной системы указанным способом учитывается, сколько схем технологического процесса не сможет выполняться, но не учитывается их стоимость.

Для производственных звеньев можно рассчитать потери, вызванные выходом из строя технологического звена  $s_i$ . Предположим, что в качестве схемы, на которую будет осуществляться переход, была выбрана схема процесса, заданная словом  $l_i$ .

**Определение 12.** Затратами производственной системы  $\bar{C}(s_i)$  относительно невозможности выполнения операции  $s_i$  назовем стоимость перехода на выбранную схему технологического процесса:  $\bar{C}(s_i) = C(l_0, l_i)$ .

Чем выше эта величина, тем большие потери вызовет невозможность выполнения операции  $s_i$ .

**Определение 13.** Долей сохраняемой стоимости систем относительно невозможности выполнения операции  $s_i$  назовем отношение стоимости замещающей схемы процесса к стоимости исходной схемы процесса:  $\Phi(s_i) = C(l_i)/C(l_0)$ .

Величина  $\Phi(s_i)$  может быть отрицательной в случае, если после возникновения аварии система работает с убытками.

На основе вероятностей отказов звеньев и долей сохраняемой стоимости введем понятие гибкости производственной системы в целом.

**Определение 14.** Гибкостью производственной системы, заданной сетью Петри  $N$ , назовем величину  $G_N =$

$$= \sum_{i=1}^S P(A_i) \Phi(s_i).$$

Эта величина характеризует способность производственной системы сохранять эффективную работу в случае возникновения ЧС на одном из ее участков. Чем она больше, тем большим структурно-технологическим резервом обладает производственная система и, как следствие, тем она устойчивее к возникающим на ней авариям в целом.

Может возникнуть вопрос о сравнении способов реагирования различных производственных систем, реализующих одну и ту же функцию, на возникающие на них аварии. Поскольку в этих системах может быть разное число операций  $s_i$ , то их сопоставление путем сравнения средних значений гибкости не показательно. Поясним это утверждение на следующем примере.

Пусть построены сети Петри  $N_1$  и  $N_2$  двух производственных систем, реализующих одну и ту же функцию. Число операций в первой  $|S_1| = 5$ , во второй —  $|S_2| = 20$ . Пусть гибкости обеих систем равны:  $G_{N_1} = G_{N_2} = 0,9$ .

Это означает, что при невозможности выполнения одной операции из строя выходит 10% технологических схем в обеих системах. Но дело в том, что невозможность выполнения одной операции означает выход из строя 20% операций в первой системе и 5% операций во второй. Следовательно, сравнивать системы с различным числом операций, основываясь только на значении средней гибкости, нельзя.

Для преодоления этого недостатка следует учитывать долю вышедших из строя операций. При невозможности выполнения одной операции эта доля составит  $1/|S|$  от общего числа операций. Доля вышедших из строя производственных схем процесса, при аварии на операции  $s_i$ , составит  $(|L| - |L_i|)/|L|$ , где  $|L|$  — число всех технологических схем процесса, выполнение которых становится невозможным при аварии на операции  $s_i$ .

Рассчитаем среднюю долю выходящих из строя технологических схем при невозможности выполнения одной операции производственной системы:

$$\begin{aligned} \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|L| - |L_i|}{|L|} &= \frac{1}{|S|} \sum_{i=1}^{|S|} \left(1 - \frac{|L_i|}{|L|}\right) = \frac{|S|}{|S|} - \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|L_i|}{|L|} = \\ &= 1 - \frac{1}{|S|} \sum_{i=1}^{|S|} G(s_i) = 1 - G_{cp}. \end{aligned}$$

Введем понятие устойчивости производственной системы.

**Определение 15.** Устойчивостью  $H$  производственной системы относительно невозможности выполнения части ее операций назовем отношение доли выходящих из строя операций к доле технологических схем процессов, выполнение которых стало невозможным.

В случае невозможности выполнения одной операции устойчивость  $H = \frac{1}{|S|(1 - G_{cp})}$ .

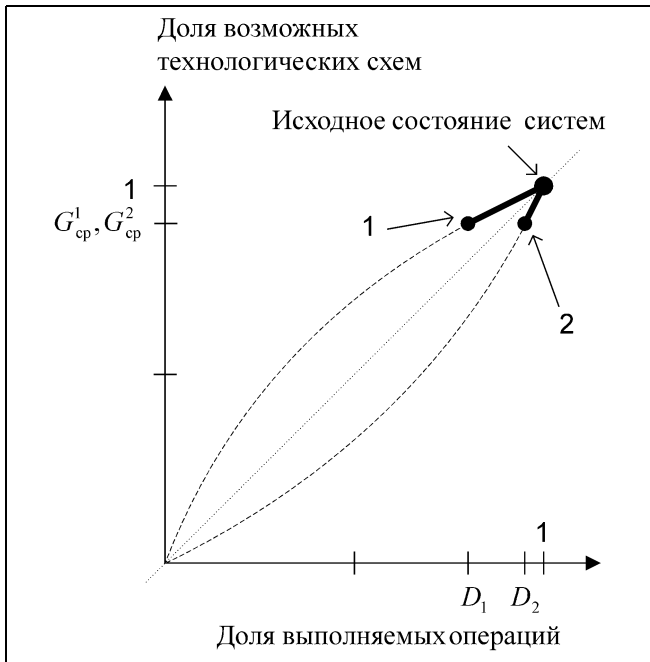


Рис. 2. Зависимость доли технологических схем процессов от доли выполняемых операций

При вычислении устойчивости учитывается доля невыполняемых операций, что позволяет сравнивать производственные системы различного масштаба.

Для приведенного выше примера устойчивость  $H_1 = \frac{1}{|S_1| \cdot (1 - G_{cp}^1)} = \frac{1}{5 \cdot 0,1} = 2$  для первой производственной системы и  $H_2 = \frac{1}{|S_2| \cdot (1 - G_{cp}^2)} = \frac{1}{20 \cdot 0,1} = 0,5$  — для второй.

Можно утверждать, что первая система сохраняет большую работоспособность при возникновении на ней аварий, поскольку  $H_1 > H_2$ .

Поясним сказанное с помощью рис. 2.

На нем изображены зависимости доли возможных технологических схем процессов от доли выполняемых операций. Точки 1 и 2 показывают состояние производственных систем при невозможности выполнения одной операции, а  $D_1$  и  $D_2$  — доли выполняемых операций, которые вычисляются как  $D_1 = (|S_1| - 1)/|S_1| = 0,8$  и  $D_2 = (|S_2| - 1)/|S_2| = 0,95$ .

Если для первой производственной системы доля выполняемых операций будет сокращаться и дальше, то доля возможных технологических схем процессов будет сокращаться не так сильно и образует верхнюю кривую, изображенную на рисунке ( $H > 1$ ). Для второй производственной системы сокращение доли выполняемых операций будет вызывать резкое снижение доли возможных технологических схем. На рисунке это будет соответствовать нижней кривой ( $H < 1$ ).

Можно сделать следующие заключения о зависимости способности производственной системы сохранять выполнимые технологические схемы процессов при возникновении аварий от значения устойчивости.

- Если значение устойчивости производственной системы ( $H < 1$ ), то при возникновении некрупных аварий и при невозможности выполнения небольшого числа операций число возможных технологических схем процессов, сохраняющихся в работоспособном состоянии, резко снижается. Производственные системы такого рода крайне чувствительны к возникающим на них авариям.
- Если значение устойчивости производственной системы ( $H > 1$ ), то при возникновении даже относительно крупных аварий и невозможности выполнения части операций число возможных технологических схем процессов остается достаточно большим. Производственные системы такого рода менее чувствительны к возникающим авариям.

## ЗАКЛЮЧЕНИЕ

Введенное понятие структурно-технологического резерва позволяет выделять совокупности операций и их технологических схем, за счет которых достигается технологическая гибкость системы, т. е. способность системы компенсировать технологические возмущения, такие как отказы и повреждения технологических звеньев, разрыв транспортных связей и т. д. С учетом характеристик производственных систем, введенных на основе структурно-технологического резерва, становится возможным выделять их наиболее уязвимые места.

В дальнейшем, на основании модели производственных систем на языке сетей Петри, могут быть разработаны формальные методы модификации производственных систем, повышающие их устойчивость.

## ЛИТЕРАТУРА

1. Архипова Н. И., Кульба В. В. Управление в чрезвычайных ситуациях. — М.: РГГУ, 1998.
2. Dimitrovici C., Hummert U., Petrucci L. Semantics, composition and net properties of algebraic high-level nets // LNCS. — 1991. — Vol. 524.
3. A unified high-level Petri net formalism for time-critical systems // C. Ghezzi, D. Mandrioli, S. Morasca, M. Pezzi // IEEE Trans. on Software Engineering. — 1991. — February.
4. Stochastic well-formed coloured nets and symmetric modelling applications // G. Chiola, C. Dutheillet, G. Franceschinis, S. Haddad // IEEE Trans. on Computers. — 1993. — Vol. 42, N 11.
5. Цвиркун А. Д., Акинфиев В. К., Филиппов В. А. Имитационное моделирование в задачах синтеза структуры сложных систем. — М.: Наука, 1985.
6. Субботин В. Е. Чрезвычайные ситуации техногенного характера. — Волгоград, Волгогр. гос. техн. ун-т, 1996.

☎ (095) 334-90-09

E-mail: kulba@ipu.ru

□





# МОДЕЛИРОВАНИЕ СМЕНЫ ТЕХНОЛОГИЧЕСКОГО УКЛАДА

В. А. Колемаев, А. Е. Бережной

Государственный университет управления, г. Москва

Поставлена задача оптимального управления процессом смены одного технологического уклада другим, получено ее решение и модельно показано, что влияние научно-технического прогресса в долгосрочном периоде, протекающее путем смены технологического уклада, при определенных условиях может приводить к образованию экономических циклов.

## ВВЕДЕНИЕ

Задача повышения уровня жизни населения, увеличения национального богатства и объемов производства, особенно актуальная в современной России, неразрывно связана с проблемой модернизации системы управления, переходом к более совершенным способам производства, к новейшим технологиям в массовом порядке. Поэтому такая задача носит не столько количественный, сколько качественный характер и сопряжена с проблемами научно-технического развития страны.

Следуя работам российских ученых С. Ю. Глазьева и Д. С. Львова [1, 2], научно-техническое развитие в долгосрочном периоде протекает путем смены технологических укладов. Под технологическим укладом понимают некоторую совокупность производств, находящихся примерно на одном и том же уровне технического развития и образующих такую экономическую систему, которая способна осуществлять процессы производства продукции и собственного расширения преимущественно за счет внутренних ресурсов.

Из концепции научно-технического прогресса как процесса смены одного технологического уклада другим естественно вытекает необходимость математического моделирования этого процесса. В настоящей статье представлена модель смены технологического уклада, поставлена задача оптимального управления процессом смены технологического уклада, получено ее решение в элементарных функциях и приведены результаты верификации модели на числовых данных.

В рамках данной работы будем различать уходящий традиционный технологический уклад, который находится на заключительном этапе своего развития, и замещающий его инновационный технологический уклад, который только начинает внедряться в экономическую систему. Будем считать, что все сбережения, на которые уменьшается непроизводительное потребление, направ-

ляются на инвестиции в производство соответствующего уклада, т. е. значения сбережений и инвестиций равны.

Инновационный технологический уклад в своем развитии проходит этап накопления и этап отдачи от накоплений, которые будут рассмотрены далее.

## 1. ЭТАП НАКОПЛЕНИЯ

На этом этапе существует всего лишь один уклад, способный осуществлять производство продукции. Мы будем придерживаться допущений, характерных для модели Солоу [3]. Согласно этой модели одноукладная экономика в долгосрочном периоде достигает своего стационарного состояния, при котором все удельные показатели постоянны, а норма накопления в традиционный уклад равна эластичности выпуска по капиталу.

Инвестиции в инновационный технологический уклад в таком случае целесообразно, на наш взгляд, осуществлять за счет уменьшения непроизводительного потребления, так как уменьшение инвестиций в традиционный уклад, который служит донором экономики страны, может негативно сказаться на экономике в случае неудачного внедрения новых технологий.

Норма накопления  $\rho_1(t)$  в инновационный технологический уклад и продолжительность  $\tau$  этапа накопления определяется, исходя из программы технико-экономического развития страны, в которой будут отражены стадии разработки необходимой проектной документации, создания производственных мощностей и инфраструктуры инновационного технологического уклада. В общем случае норма накопления может быть переменной.

Таким образом, за  $\tau$  лет общий объем инвестиций в инновационный технологический уклад  $K_1^1 = \int_0^{\tau} \eta \rho_1(t) X_0 dt$ .

Параметр  $\eta$  характеризует неизбежные потери, связанные с использованием старого оборудования для создания техники инновационного уклада.

**Основные обозначения:**

Экзогенные переменные

 $L$  — численность рабочей силы во всей экономике

 $\mu_0, \mu_1$  — нормы выбытия фондов

 $\rho_0, \rho_1$  — нормы накопления укладов

 $\rho_1^{\min}, \rho_1^{\max}$  — пороговые значения нормы накопления в инновационный уклад

 $w_1$  — ставка заработной платы

 $p_0$  — цена единицы продукции традиционного уклада

 $\tau$  — длина этапа накопления

Эндогенные переменные

 $X_0, X_1$  — валовые выпуски укладов<sup>1</sup>
 $L_0, L_1$  — численность рабочей силы укладов

 $K_0, K_1$  — объемы основных фондов укладов

 $C$  — непроемленное потребление

 $w_0$  — ставка заработной платы традиционного уклада

 $k_0 = K_0/L_0, k_1 = K_1/L_1$  — фондовооруженность укладов

 $\theta_0 = L_0/L, \theta_1 = L_1/L$  — доли работающих в укладах

 $c$  — удельное непроемленное потребление

 $x_0 = X_0/L_0, x_1 = X_1/L_1$  — удельные выпуски укладов в расчете на одного занятого

 $T$  — длина этапа отдачи от накоплений

 $\tilde{T} = T + \tau$  — момент окончания этапа отдачи от накоплений

 $t^*$  — момент переключения управления, т. е. перехода нормы накопления с одного экстремального значения на другое.

 Индексом “0” будем обозначать показатели, относящиеся к традиционному укладу, “1” — к инновационному. Верхним индексом “E” будем обозначать стационарные показатели, верхним индексом “1” — начальные условия. Так, например, стационарную и начальную фондовооруженности традиционного уклада будем соответственно обозначать  $k_0^E$  и  $k_0^1$ .

**2. ЭТАП ОТДАЧИ ОТ НАКОПЛЕНИЙ**

На рассматриваемом этапе лаг капиталовложений в инновационный уклад отсутствует. Связь выпуска традиционного уклада с затратами труда и капитала будем считать подчиненной зависимости, выражаемой с помощью производственной функции Кобба—Дугласа:

$$X_0 = A_0 K_0^{\alpha_0} L_0^{1-\alpha_0}. \quad (1)$$

В связи с тем, что на начальном этапе появления новой технологии наблюдается постоянная отдача от увеличения использования факторов производства, которая связана напрямую с обучением рабочей силы в производственном процессе, для отражения взаимосвязи объемов выпуска с затратами труда и капитала будем пользоваться производственной функцией Леонтьева.

<sup>1</sup> В момент времени  $t$ . Здесь и далее, кроме случаев, когда это особенно необходимо подчеркнуть, для простоты аргумент  $t$  будет опускаться.

Такая производственная функция более адекватно, чем линейная, отражает производственный процесс, так как последняя допускает абсолютную взаимозаменяемость между трудом и капиталом, что противоречит тому факту, что именно капитал в виде материализованных инноваций является носителем новых технологий.

Другие, широко распространенные в экономическом анализе производственные функции, такие как постоянная эластичности замещения, в том числе производственная функция Кобба—Дугласа, также менее предпочтительны для описания поведения инновационного уклада на рассматриваемом этапе, так как для них характерна убывающая отдача от увеличения использования какого-либо фактора производства.

Будем считать, что в силу большей привлекательности инновационного уклада (более высокая оплата труда, новое оборудование, перспективность работы в высокотехнологичном секторе, лучшие экологические условия и т. п.), рабочая сила будет избыточным фактором, а ограничивающим расширение новых технологий фактором будет капитал (в том смысле, что  $\alpha_K K_1 \leq \alpha_L L_1$ ).

Поэтому производственная функция в таком случае примет следующий вид:

$$X_1 = \min(a_K K_1, a_L L_1) = a_K K_1, \quad (2)$$

где  $a_K$  и  $a_L$  — некоторые неотрицательные коэффициенты. Численность рабочей силы будем считать неизменной:

$$L = L_1 + L_0 = \text{const}. \quad (3)$$

Кроме того, упомянутая привлекательность будет служить мотивом перехода рабочей силы из традиционного уклада в инновационный. Скорость этого перехода будет напрямую зависеть от соотношения ставок заработной платы в традиционном и инновационном укладах, поэтому вместе с учетом соотношения (3) получаем:

$$-\frac{dL_0}{dt} = \frac{dL_1}{dt} = \beta \frac{w_1}{w_0}, \quad (4)$$

где  $\beta$  — коэффициент, характеризующий скорость перераспределения трудовых ресурсов из традиционного уклада в инновационный при равных ставках заработной платы, он показывает, на сколько уменьшится численность рабочей силы традиционного уклада в единицу времени, если ставки заработных плат укладов будут равны.

Чем больше ставка заработной платы, тем меньше будет желающих сменить место работы. Напомним, что в экономической теории ставку заработной платы принято считать в долгосрочном периоде равной стоимости предельного продукта на одного занятого. Ставка заработной платы для работников инновационного уклада в таком случае будет постоянна.

Предельный продукт на одного занятого в традиционном укладе выражается как  $\frac{\partial X_0}{\partial L_0} = A_0(1 - \alpha_0)k_0^{\alpha_0}$ , где  $A_0$  и  $\alpha_0$  — неотрицательные константы.

Однако на практике ставку заработной платы часто занижают по сравнению со стоимостью предельного продукта, поэтому для традиционного уклада:

$$w_0 = \zeta p_0 \frac{\partial X_0}{\partial L_0} = \zeta p_0 A_0(1 - \alpha_0)k_0^{\alpha_0},$$

где  $\zeta, 0 < \zeta < 1$ , — коэффициент занижения.



Отсюда получим:

$$\frac{dL_0}{dt} = -\Gamma \frac{w_1}{A_0(1-\alpha_0)k_0^{\alpha_0}}, \quad L_0(\tau) = L_0^1, \quad \Gamma = \frac{\beta}{\zeta p_0}. \quad (5)$$

Из выражений (3) и (5) нетрудно получить:

$$\frac{dL_1}{dt} = \Gamma \frac{w_1}{A_0(1-\alpha_0)k_0^{\alpha_0}}, \quad L_1(\tau) = L_1^1. \quad (6)$$

Ввиду высокой эффективности инновационного уклада будем полагать, что все инвестиции целесообразно направлять в этот уклад:

$$\frac{dK_0}{dt} = -\mu_0 K_0, \quad K_0(\tau) = K_0^1, \quad (7)$$

$$\frac{dK_1}{dt} = -\mu_1 K_1 + \rho_1(t)(X_1 + \eta X_0), \quad K_1(\tau) = K_1^1. \quad (8)$$

Параметр  $\eta$ ,  $0 < \eta < 1$ , как и прежде, отражает тот факт, что на новом оборудовании процесс воспроизводства фондов инновационного уклада происходит эффективнее, чем на старом, и без дополнительных потерь. Он также служит и для сопоставления продукции укладов по качеству с точки зрения непроизводственного потребления, т. е.  $X_0$  единиц продукции традиционного уклада эквивалентны для потребителя  $X_1$  единиц продукции инновационного уклада, если  $X_1 = \eta X_0$ .

Рассмотрим теперь вопрос о значениях величин  $K_0$  и  $K_1$  в момент  $\tau$ . Значение  $K_1^1$  определяется на предыдущем этапе развития уклада. В начальный момент времени для запуска производства будет нанято некоторое число людей. Управляющий орган будет стремиться к тому, чтобы рабочая сила не была, с одной стороны, ограничивающим фактором, а с другой — ее не должно быть чрезмерно много. Иными словами, должно выполняться равенство

$$X_1^1 = \min(a_K K_1^1, a_L L_1^1) = a_K K_1^1 = a_L L_1^1 \Rightarrow L_1^1 = a_K K_1^1 / a_L. \quad (9)$$

Из соотношений (3) и (9) получаем  $L_0^1 = L - a_K K_1^1 / a_L$ . В момент времени  $\tau$  размер основных фондов традиционного уклада  $K_0^1 = L k_0^E$ .

Поставим теперь задачу оптимального управления [4] процессом смены технологического уклада. В качестве управляющего параметра в данном случае будет выступать норма накопления  $\rho_1(t)$ . Отметим, что варьировать нормой накопления в инновационный уклад можно не только в экономической системе с плановой экономикой. В экономической системе с рыночной экономикой у государства есть целый ряд гибких рыночных инструментов для регулирования этого параметра. В России таким инструментом может выступать ставка рефинансирования Центрального Банка.

Роль фазовых переменных будут выполнять фондовооруженность укладов и доля занятых в традиционном укладе, траектории их движения описываются нижеприведенной системой уравнений (11), функционалом бу-

дет выступать функция интегрального потребления за рассматриваемый период  $T$ , которую необходимо максимизировать:

$$\int_{\tau}^{T+\tau} (1 - \rho_1(t))(X_1 + \eta X_0) dt + \delta K_1(T) \xrightarrow{\rho_1(t) \in [\rho_1^{\min}, \rho_1^{\max}]} \max, \quad (10)$$

где  $\delta$ ,  $0 \leq \delta \leq 1$ , — заранее заданная константа, характеризующая наше предпочтение настоящего будущему.

Запишем функцию (10) в относительных показателях, поделив ее на константу  $L$ :

$$\int_{\tau}^{T+\tau} (1 - \rho_1(t))(x_1(1 - \theta_0) + \eta x_0 \theta_0) dt + \delta k_1(T) \xrightarrow{\rho_1(t) \in [\rho_1^{\min}, \rho_1^{\max}]} \max.$$

Уравнения эволюции системы (1)—(8) нетрудно преобразовать в систему нелинейных дифференциальных уравнений относительных показателей:

$$\begin{cases} \frac{dk_0}{dt} = -\mu_0 k_0 + k_0 \frac{b_0}{k_0^{\alpha_0} \theta_0}, & k_0(\tau) = k_0^1 \\ \frac{dk_1}{dt} = \rho_1(t) \left( x_1 + \frac{\theta_0}{1 - \theta_0} \eta x_0 \right) - k_1 \frac{b_0}{(1 - \theta_0) k_0^{\alpha_0}} - \mu_1 k_1, & k_1(\tau) = k_1^1 \\ \frac{d\theta_0}{dt} = -\frac{b_0}{k_0^{\alpha_0}}, & \theta_0(\tau) = \theta_0^1 \end{cases} \quad (11)$$

где  $x_0 = X_0/L_0 = A_0 k_0^{\alpha_0}$  и  $x_1 = X_1/L_1 = a_k k_1$  — производительности труда традиционного и инновационного укладов,  $\gamma = \Gamma/L$  и  $b_0 = \frac{\gamma w_1}{A_0(1-\alpha_0)}$  — константы.

Начальные условия можно записать в виде:

$$\begin{cases} k_1^1 = \frac{a_1}{a_k} \\ \theta_0^1 = 1 - \frac{K_1^1}{k_1^1 L} = 1 - \frac{a_K K_1^1}{a_L L} \\ k_0^1 = k_0^E \frac{1}{\theta_0^1} = k_0^E \frac{1}{1 - \frac{a_K K_1^1}{a_L L}} \end{cases}$$

Для облегчения последующего решения задачи сдвинем начало координат по оси времени из точки  $\tau$  в точку 0. Перепишем начальные условия в новых координатах:

$$\begin{cases} k_1(0) = k_1^1 \\ k_0(0) = k_0^1 \\ \theta_0(0) = \theta_0^1 \end{cases}$$

Условия трансверсальности для сопряженных функций зададим следующим образом [5]:  $\psi_1(T) = \frac{\partial(\delta k_1(T))}{\partial k_0}$ ,

$$\psi_2(T) = \frac{\partial(\delta k_1(T))}{\partial k_1}, \quad \psi_3(T) = \frac{\partial(\delta k_1(T))}{\partial \theta_0}, \quad \text{откуда}$$

$$\psi_1(T) = 0, \quad \psi_2(T) = \delta, \quad \psi_3(T) = 0. \quad (12)$$

Гамильтонова функция примет вид:

$$\begin{aligned} H = & \psi_0[(1 - \rho_1)(x_1(1 - \theta_0) + \eta x_0 \theta_0)] + \\ & + \psi_1(t) \left[ k_0 \frac{\gamma w_1}{A_0(1 - \alpha_0) k_0^{\alpha_0} \theta_0} - \mu_0 k_0 \right] + \\ & + \psi_2(t) \left[ \rho_1 \left( x_1 + \frac{\theta_0}{1 - \theta_0} \eta x_0 \right) - k_1 \frac{\gamma w_1}{A_0(1 - \alpha_0)(1 - \theta_0) k_0^{\alpha_0}} - \right. \\ & \left. - \mu_1 k_1 \right] - \psi_3(t) \frac{\gamma w_1}{A_0(1 - \alpha_0) k_0^{\alpha_0}}. \end{aligned}$$

Как видно, она линейна относительно управляющего параметра  $\rho_1(t)$ , следовательно, он будет принимать только свои экстремальные значения, в зависимости от знака ее производной по этому параметру:

$$\begin{cases} \rho_1 = \rho_1^{\max}, \frac{\partial H}{\partial \rho_1} > 0 \\ \rho_1 = \rho_1^{\min}, \frac{\partial H}{\partial \rho_1} < 0 \\ \rho_1 \text{ изменяет свое значение, } \frac{\partial H}{\partial \rho_1} = 0. \end{cases} \quad (13)$$

Таким образом,  $\rho_1(t) = \rho_1 = \text{const}$  для всех  $t$ , при которых  $\frac{\partial H}{\partial \rho_1} \neq 0$  и только когда  $\frac{\partial H}{\partial \rho_1} = 0$ , управляющий параметр скачкообразно изменяет свое значение, т. е.  $\rho_1(t)$  — ограниченная кусочно-непрерывная справа функция времени.

Приведем строгую постановку задачи оптимального управления:

$$\int_0^T (1 - \rho_1)(x_1(1 - \theta_0) + \eta x_0 \theta_0) dt + \delta k_1(T) \xrightarrow{\rho_1 \in [\rho_1^{\min}, \rho_1^{\max}]} \max.$$

Уравнения эволюции системы:

$$\begin{cases} \frac{dk_0}{dt} = -\mu_0 k_0 + k_0 \frac{b_0}{k_0^{\alpha_0} \theta_0}, \quad k_0(0) = k_0^1 \\ \frac{dk_1}{dt} = \rho_1 \left( x_1 + \frac{\theta_0}{1 - \theta_0} \eta x_0 \right) - k_1 \frac{b_0}{(1 - \theta_0) k_0^{\alpha_0}} - \mu_1 k_1, \\ \quad k_1(0) = k_1^1 \\ \frac{d\theta_0}{dt} = -\frac{b_0}{k_0^{\alpha_0}}, \quad \theta_0(0) = \theta_0^1, \end{cases} \quad (14)$$

Сопряженная система:

$$\begin{cases} \frac{d\psi_1}{dt} = -\frac{\partial H}{\partial k_0}, \quad \psi_1(T) = 0 \\ \frac{d\psi_2}{dt} = -\frac{\partial H}{\partial k_1}, \quad \psi_2(T) = \delta \\ \frac{d\psi_3}{dt} = -\frac{\partial H}{\partial \theta_0}, \quad \psi_3(T) = 0. \end{cases} \quad (15)$$

Начальные условия системы (14) принадлежат некоторой замкнутой области  $D : \{\theta_0 \in [\underline{\theta}_0; \bar{\theta}_0] \subset (0; 1), k_0 \in [k_0; \bar{k}_0] \subset (0; \infty), k_1 \in [k_1; \bar{k}_1] \subset (0; \infty)\}$ , причем правые части дифференциальных уравнений этой системы непрерывны и ограничены вместе со своими частными производными на области  $D$ . Отсюда следует, что решение системы (14) существует и единственно.

Решим теперь систему (14). Поделив каждую часть первых двух ее уравнений на соответствующие части третьего уравнения, получим:

$$\begin{cases} \frac{dk_0}{d\theta_0} + \frac{k_0}{\theta_0} = \frac{\mu_0 A_0 (1 - \alpha_0) k_0^{\alpha_0 + 1}}{\gamma w_1} \\ \frac{dk_1}{d\theta_0} + \left[ \frac{(\rho_1 a_k - \mu_1) A_0 (1 - \alpha_0) k_0^{\alpha_0}}{\gamma w_1} - \frac{1}{1 - \theta_0} \right] k_1 = \\ = -\rho_1 \frac{\theta_0}{1 - \theta_0} \frac{A_0^2 (1 - \alpha_0) \eta}{\gamma w_1} k_0^{2\alpha_0}. \end{cases} \quad (16)$$

Общее решение первого уравнения (уравнения Бернулли) системы (16):

$$\begin{aligned} k_0^{-\alpha_0} &= -\frac{\alpha_0 \mu_0 A_0}{\gamma w_1} \theta_0 + B_0 \theta_0^{\alpha_0}, \\ k_0 &= \left( B_0 \theta_0^{\alpha_0} - \frac{\alpha_0 \mu_0 A_0}{\gamma w_1} \theta_0 \right)^{\frac{1}{\alpha_0}}, \end{aligned} \quad (17)$$

где  $B_0$  — константа, определяемая из начальных условий; можно показать, что  $B_0 > 0$ .

Если решение (17) подставим во второе уравнение системы (16), то получим следующее решение:

$$\begin{aligned} k_1 &= \rho_1 \frac{\gamma w_1 \eta}{\alpha_0 \mu_0} \left[ \frac{1}{(\rho_1 \alpha_k - \mu_1)(1 - \theta_0)} - \right. \\ & \left. - \frac{B_0 \theta_0^{\alpha_0}}{(\rho_1 \alpha_k - \mu_1 + \alpha_0 \mu_0)(1 - \theta_0) \left( B_0 \theta_0^{\alpha_0} - \frac{\alpha_0 \mu_0 A_0}{\gamma w_1} \theta_0 \right)} \right] + \\ & + \frac{B_1}{1 - \theta_0} \left| \frac{B_0 \theta_0^{\alpha_0}}{B_0 \theta_0^{\alpha_0} - \frac{\alpha_0 \mu_0 A_0}{\gamma w_1} \theta_0} \right|^{\frac{\mu_1 - \rho_1 \alpha_k}{\alpha_0 \mu_0}}. \end{aligned}$$

Здесь  $B_1$  также некоторая константа при фиксированном  $\rho_1$ , но так как  $\rho_1$  — варьируемый параметр, то  $B_1 = B_1(\rho_1)$ .





Решение третьего уравнения системы (14):

$$\theta_0 = \left[ B_0 e^{\alpha_0 \mu_0 t} + \frac{\gamma w_1 B_0}{A_0 \alpha_0 \mu_0} \right]^{\frac{1}{1-\alpha_0}}, \quad (18)$$

где  $B_0$  — константа, определяемая из начальных условий; можно показать, что  $B_0 < 0$ .

По окончании этапа отдачи от накоплений инновационный технологический уклад полностью вытеснит традиционный, т. е.  $\theta_0(T) = \left[ B_0 e^{\alpha_0 \mu_0 T} + \frac{\gamma w_1 B_0}{A_0 \alpha_0 \mu_0} \right]^{\frac{1}{1-\alpha_0}} = 0$ , откуда

$$T = \frac{1}{\alpha_0 \mu_0} \ln \left[ -\frac{\gamma w_1 B_0}{A_0 \alpha_0 \mu_0 B_0} \right]. \quad (19)$$

Найдем теперь момент переключения. Согласно условиям (13), его можно найти из уравнения:

$$\begin{aligned} \frac{\partial H}{\partial \rho_1} = & -\psi_0 [x_1(1-\theta_0) + \eta x_0 \theta_0] + \\ & + \left( x_1 + \frac{\theta_0}{1-\theta_0} \eta x_0 \right) \psi_2(t) = 0. \end{aligned} \quad (20)$$

Константу  $\psi_0 \geq 0$  принимают равной единице для задачи на максимум, а функция  $\psi_2(t)$  определяется в результате решения соответствующего уравнения сопряженной системы (15), т. е.  $\frac{d\psi_2}{dt} = -\frac{\partial H}{\partial k_1} = -\psi_0 a_k (1-\theta_0) \times (1-\rho_1) + \psi_2(t) \left[ \frac{\gamma w_1}{A_0(1-\alpha_0)(1-\theta_0)k_0^{\alpha_0}} + \mu_1 - \rho_1 a_k \right]$ .

Поделим это уравнение на третье уравнение системы (14):

$$\begin{aligned} \frac{d\psi_2}{d\theta_0} = & \frac{\psi_0 a_k (1-\theta_0)(1-\rho_1)}{\gamma w_1} A_0 (1-\alpha_0) k_0^{\alpha_0} + \\ & + \psi_2 \left[ \frac{\rho_1 a_k - \mu_1}{\gamma w_1} A_0 (1-\alpha_0) k_0^{\alpha_0} - \frac{1}{1-\theta_0} \right]. \end{aligned}$$

Решение полученного уравнения имеет вид:

$$\begin{aligned} \psi_2(\theta_0) = & \frac{-\psi_0 a_k (1-\rho_1)}{\rho_1 a_k - \mu_1} (1-\theta_0) + \\ & + B_\psi \left| \frac{B_0 \theta_0^{\alpha_0}}{B_0 \theta_0^{\alpha_0} - \frac{\alpha_0 \mu_0 A_0}{\gamma w_1} \theta_0} \right|^{\frac{\rho_1 a_k - \mu_1}{\alpha_0 \mu_0}} (1-\theta_0). \end{aligned} \quad (21)$$

Константа  $B_\psi$  определяется из условия (12).

Экономическое содержание двух слагаемых выражения (20) состоит в следующем: первое слагаемое характеризует ценность для общества увеличения потребления в момент времени  $t$  с отрицательным знаком, так как чем больше эта полезность, тем меньше должны быть сбережения (напомним, что в рамках настоящей работы инвестиции и сбережения приняты равными). Второе слагаемое неотрицательно и характеризует общественную полезность сбережений в текущий момент

времени. Чем выше эта полезность, тем выше должна быть норма накопления. Отсюда трактовка функции  $\psi_2(t)$  как теневой цены прироста фондов.

Из уравнения (20) получим:

$$\psi_2(t^*) = \psi_2(\theta_0^*) \psi_0 (1-\theta_0^*), \quad \theta_0^* = \theta_0(t^*) \quad (22)$$

Подставив в выражение (22) функцию (21), получим

$$\theta_0^* = \left\{ \frac{\gamma w_1 B_0}{\alpha_0 \mu_0 A_0} \left[ 1 - \left( \frac{(\mu_1 - a_k) \psi_0}{B_\psi (\mu_1 - \rho_1 a_k)} \right)^{\frac{\alpha_0 \mu_0}{\mu_1 - \rho_1 a_k}} \right] \right\}^{\frac{1}{1-\alpha_0}}. \quad (23)$$

Разрешив соотношение (18) относительно  $t$  и подставив в полученное выражение формулу (23), найдем момент переключения

$$\begin{aligned} t^* = & \frac{1}{\alpha_0 \mu_0} \ln \left( -\frac{\gamma w_1 B_0}{B_0 \alpha_0 \mu_0 A_0} \right) - \\ & - \frac{1}{\rho_1 a_k - \mu_1} \ln \left( \frac{(\mu_1 - a_k) \psi_0}{B_\psi (\rho_1 a_k - \mu_1)} \right). \end{aligned} \quad (24)$$

В выражении для  $t^*$  первое слагаемое, согласно формуле (19), — это период  $T$ , а второе слагаемое равно  $(-ΔT)$ , где  $ΔT = \frac{1}{\rho_1 a_k - \mu_1} \ln \left( \frac{(\mu_1 - a_k) \psi_0}{\psi_0 a_k + (\delta - \psi_0) a_k \rho_1 - \delta \mu_1} \right)$ .

Величина  $ΔT$  — убывающая функция от  $\delta$ . Если  $\delta = 1$ , то  $ΔT = 0$  и  $t^* = T$ , поэтому в течение всего планируемого периода  $T$  управляющий параметр будет принимать только одно экстремальное значение.

При других значениях  $\delta$  на отрезке от 0 до 1  $ΔT > 0$ , а управляющий параметр может принимать как минимальное, так и максимальное значения.

Можно также показать, что момент переключения  $t^*$  единственен, а управляющий параметр будет принимать вначале наибольшее значение, а затем, в случае  $t^* < T$ , наименьшее; т. е. управление будет носить следующий характер: вначале норма накопления максимальна, затем в момент  $t^*$ , определяемый выражением (24), ее значение переключается на минимальное.

Учитывая единственность решения системы (14), можно утверждать, что найденное управление, удовлетворяющее принципу максимума Понтрягина, оптимально.

### 3. ВЕРИФИКАЦИЯ МОДЕЛИ НА ЧИСЛОВЫХ ДАННЫХ

На числовых данных была проведена верификация полученных результатов. Производственная функция Кобба—Дугласа соответствовала производственной функции России с 1970 по 1990 г., начальные условия — статистическим данным по России за 2000 г. Значения параметра  $\gamma$ , характеризующего скорость перераспределения рабочей силы из уклада в уклад, варьировались.

Интересна экономическая интерпретация параметра  $\gamma$ . С ним связан функциональной зависимостью параметр  $\beta/L = \gamma \zeta$ , который выражает долю рабочей силы, перешедшей из традиционного уклада в инновационный уклад в течение года, если ставки заработной платы укладов были равны. Естественно, что при равных ставках заработных плат из уклада в уклад будут переходить

**Основные результаты моделирования на числовых данных**

$T$	$t^*$	$\gamma$	Доля инноваторов $\beta/L = \gamma\zeta$	Динамика удельного потребления
4,4	3,1	1,1	0,286	Спад
4,8	3,5	1,0	0,260	Цикл
5,3	4,0	0,9	0,234	Рост

инноваторы<sup>2</sup>, тогда как консерваторы предпочтут остаться в своем укладе. Таким образом,  $\beta/L$  — показатель инновационной активности в экономике, т. е. как бы доля инноваторов среди занятых.

В результате верификации оказалось, что динамика удельного непроизводственного потребления в зависимости от доли инноваторов в экономике может принимать циклический характер (начальный спад сменяется последующим ростом), монотонно возрастающий и убывающий характер в течение всего периода этапа отдачи от накоплений. Таким образом, было модельно показано, что научно-технический прогресс в рамках процесса смены технологического уклада может быть источником экономических циклов (см. таблицу).

Отметим, что, несмотря на небольшую продолжительность периода  $T$  вытеснения традиционного уклада, речь идет о длинных волнах Кондратьева, так как после смены традиционного уклада будет длительный период господства инновационного уклада, вытеснение которого и завершит этот цикл; т. е. продолжительность цикла следует измерять с момента возникновения уклада до момента его вытеснения.

**ЗАКЛЮЧЕНИЕ**

В рамках проведенной работы было показано, что управление перевооружением экономики должно носить релейный характер: вначале норма накопления принимает максимальное значение, затем в момент (24) она должна быть переключена на минимальное значение.

<sup>2</sup> Под инноваторами здесь понимается та часть рабочей силы, которая согласна сменить свое место работы в традиционном технологическом укладе на новое в инновационном технологическом укладе при равных ставках заработной платы в укладах в течение года. Консерваторы, в отличие от инноваторов, несклонны к такой смене места работы.

Процесс смены технологического уклада может носить циклический, равномерный и убывающий характер в зависимости от перераспределения ресурсов из одного уклада в другой. При плавном, постепенном перераспределении рабочей силы из уклада в уклад наблюдается равномерность процесса, однако излишняя инертность кадров может сильно растянуть этот процесс во времени, занизив при этом темпы научно-технического развития. Резкий перелив ресурсов в инновационный уклад, что часто бывает на практике, может привести к цикличности процесса, т. е. вначале будет спад, а затем подъем, или к обвальному падению удельного непроизводственного потребления.

Равномерность перераспределения рабочей силы в рамках предложенной модели зависит от двух составляющих: отношения ставок заработной платы укладов и от доли инноваторов в экономике. Расчеты показали, что если число инноваторов среди рабочей силы превышает некоторый пороговый уровень, то тогда прогресс будет носить циклический характер. Увеличение числа инноваторов сверх этого уровня может привести к кризису. Снижение — к равномерному ходу научно-технического прогресса, но при увеличенных сроках. Здесь, на наш взгляд, важной является стабилизирующая роль консерваторов.

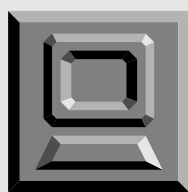
Таким образом, модельно продемонстрировано, что научно-технический прогресс в долгосрочном периоде может приводить к появлению экономических циклов, а при некоторых условиях и к экономическим кризисам, что указывает на необходимость государственного воздействия на этот процесс.

**ЛИТЕРАТУРА**

1. Глазьев С. Ю. Теория долгосрочного технико-экономического развития. — М.: ВладДар, 1993. — 310 с.
2. Глазьев С. Ю. Экономическая теория технического развития. — М.: Наука, 1990. — 232 с.
3. Колмаев В. А. Математическая экономика. — М.: ЮНИТИ, 1998. — 240 с.
4. Математическая теория оптимальных процессов / Л. С. Понтрягин, В. Г. Болтянский, Р. В. Гамкрелидзе, Е. Ф. Мищенко. — М.: Наука, 1983. — 392 с.
5. Афанасьев В. Н., Колмановский В. Б., Носов В. Р. Математическая теория конструирования систем управления. — М.: Высш. шк., 1998. — С. 176.

☎ (095) 371-11-65

E-mail: [al\\_berej@pochta.ru](mailto:al_berej@pochta.ru)

**Новая электронная гиперкнига**

**В.А. Трапезников "Управление и научно-технический прогресс"**. Книга выдающегося российского ученого, академика Вадима Александровича Трапезникова — ценный источник знаний для руководителей, менеджеров, экономистов, инженеров, преподавателей и ученых, интересующихся вопросами эффективных инноваций в сфере управления предприятиями и организациями.

В книге рассматриваются вопросы: управление как источник прогресса, стратегия управления, вопросы управления экономическими системами, человек в системе управления, стимулы прогресса, автоматизация как основная форма научно-технического прогресса, непосредственно связанная с управлением и др.

Огромная научная эрудиция, государственное мышление, способность предвидеть развитие науки и техники позволили В.А. Трапезникову не только внести выдающийся вклад в становление и организацию науки управления у нас в стране, но и оставить нам богатейшее научное наследие, которое еще предстоит осмыслить и использовать.

Книгу можно получить бесплатно, обратившись по адресу [epstein@ipu.rssi.ru](mailto:epstein@ipu.rssi.ru)



# УПРАВЛЕНИЕ ДИВЕРСИФИКАЦИЕЙ ДЕЯТЕЛЬНОСТИ ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ НА РЕГИОНАЛЬНОМ РЫНКЕ

О. А. Лузгина

*Пензенский государственный университет*

Рассмотрена целесообразность управления диверсификацией деятельности промышленных предприятий на региональном рынке. Выполнен анализ портфелей деятельности предприятий машиностроительного комплекса Пензенской области, выявивший имеющийся потенциал развития диверсификации. Предложены мероприятия в управлении процессами диверсификации, предполагающие исследование интересов субъектов и включение ресурса исполнительной власти в регионе.

Намеченный руководством страны рост объемов валового внутреннего продукта России вызывает необходимость пересмотра стратегических задач и планов промышленных предприятий в регионах страны. В то же время глобализация экономики значительно расширяет возможности интеграции российских предприятий в сферы мирового бизнеса, при этом увеличивается вероятность свертывания уже занятых рыночных ниш. Эти факторы требуют активизации менеджмента промышленных предприятий в направлении диверсификации.

К этой проблеме следует добавить и необходимость решения социально-экономических задач региона, в котором располагается промышленное предприятие: создание дополнительных рабочих мест, наполняемость бюджетов. Необходимо также учитывать, что в данный момент делается ставка на развитие экономики регионов в целях повышения степени их самообеспеченности и роста эффективности их экономических систем. Исходя из стратегии развития региона и потребностей субъектов региональной экономики в интенсификации хозяйственной жизни, появляются новые сегменты рынков, на которых могут успешно работать функционирующие в регионе предприятия.

Диверсификация промышленного предприятия осуществляется на базе имеющихся сфер деятельности и, в первую очередь, используется информация о продуктовом дефиците регионального рынка, для покрытия которого возможно привлечение потенциала расположенных в том же регионе хозяйствующих субъектов. Цель настоящей работы заключается в анализе возможности диверсификации деятельности промышленных предприятий на региональном рынке с перспективой поиска организационно-экономического механизма управления этим процессом.

В нынешних условиях хозяйствования, когда большинству предприятий и организаций предоставлена хозяйственная самостоятельность, реструктуризирован ряд отраслевых министерств (за исключением естественных монополий), все труднее обеспечить макроэкономическое единство отраслевых региональных комплексов. Отметим также, что рыночные отношения, развитие правовой базы свободного предпринимательства в России ограничивают возможности влияния региональной власти на характер и логику развития наиболее перспективных отраслей экономики.

Выделим факторы динамики регионального рынка, определяющие возможности управления процессами диверсификации:

- административное деление страны на регионы, каждый из которых обладает собственным экономическим потенциалом;
- наличие в регионах властных структур, в том числе территориальных органов управления;
- формирование бюджетов региона за счет налоговых отчислений хозяйствующих субъектов, находящихся на территории региона;
- наличие региональных финансовых структур — банков, страховых компаний, инвестиционных фондов;
- наличие региональных общественных организаций поддержки предпринимательства и промышленных товаропроизводителей;
- наличие на территории региона групп товаропроизводителей и покупателей, имеющих общие хозяйственные интересы;
- возможность использования ресурсов местного территориального рынка труда и образовательных услуг;
- формирование территориальной информационной базы по спросу и предложению отдельных товаров;

— возможность формирования координационных территориальных органов управления;

— наличие территориальной инфраструктуры рынка (транспорта, энергоснабжения, торговой сети, системы медицинского и социального обеспечения).

Наличие уникальных природных ресурсов, дешевой и (или) квалифицированной рабочей силы, производственных мощностей соответствующих групп отраслей снижает производственные затраты в том или ином регионе и позволяет фирме конкурировать за счет низких затрат, снизить транзакционные издержки. Появление в регионе, имеющем благоприятные факторы производства, нескольких фирм-конкурентов приведет к усилению соперничества или, возможно, сотрудничества между ними и, как следствие, укреплению их лидирующей позиции в отрасли. Местный рынок квалифицированной рабочей силы, переносимой «ноу-хау» с одного предприятия на другое, будет способствовать быстрому внедрению нововведений. Закрепление этой ситуации усилит внимание региональных властей к основным налогоплательщикам, что повлечет создание для них благоприятной инфраструктуры и содействие в инновационной активности.

Эти же факторы и конкурентные преимущества предприятий в отдельно взятом регионе позволяют использовать преимущества, планировать и развивать диверсификацию деятельности внутри отраслевого регионального рынка. Диверсификация может затронуть как горизонтальный уровень производственного цикла, так и носить вертикальный характер; может быть связанной и не связанной.

Автор исходит из предположения, что, несмотря на существование общих принципов управления предприятиями, имеются различия, определяемые отраслевой спецификой, целями предприятия, конкурентной средой, социально-культурными и экономическими особенностями региона. Существуют и факторы территориального расположения, оказывающие существенное влияние на формирование стратегии диверсификации деятельности предприятий, образующих отраслевые комплексы в регионе. Иными словами, возможно формирование кластер-проектов диверсификации с учетом приоритетов развития экономики региона. Под кластер-проектом в данном случае понимается обособленная часть отрасли, условно ограниченная территориальными признаками, характеризующаяся институционализацией отношений предприятий и организаций, участвующих в реализации инновационного проекта. Так, например, по предложению губернатора Пензенской области для реализации социально-экономической программы поддержки сельских районов при совместном участии нескольких промышленных предприятий были разработаны и изготовлены передвижные стоматологические комплексы.

Перечислим основные предпосылки принятия решения о диверсификации деятельности отдельных предприятий с учетом регионального фактора:

— конкурентные преимущества, которыми располагает предприятие для реализации стратегии диверсификации на региональном рынке;

— наличие доступных форм проникновения и действия на региональном рынке, завоевания и удержания рыночных ниш;

— возможность формирования (изменения) концепции управления предприятием и ее отражение и уточнение в региональном масштабе, а также в частных стратегиях активности предприятия (финансовой, сбытовой, НИОКР и т. п.).

При анализе стратегий предприятий, их структуры и деловой среды, существует возможность формулирования вопросов, позволяющих выбрать направления диверсификации наилучшим образом:

— соответствуют ли стиль менеджмента и преобладающие типы организационных структур предприятий потребностям отрасли в регионе;

— какие виды стратегии предприятий основаны на принятых в регионе организационных нормах;

— привлекает ли отрасль в регионе квалифицированных специалистов;

— способствуют ли цели инвесторов потребностям увеличения конкурентоспособности отрасли;

— каков тип отраслевого рынка?

При анализе факторов производства:

— располагает ли регион особенно развитыми или наиболее подходящими факторами производства;

— располагает ли регион механизмами создания факторов производства (ресурсов) для данной отрасли (специальные исследовательские программы, учебные заведения и т. п.);

— являются ли проблемы ресурсного обеспечения соответствующих отраслей отражением общей макроэкономической ситуации развития страны или следствием депрессивной экономической динамики региона;

— в какой мере институциональная среда региона соответствует потребностям развития данной отрасли?

Специфика спроса в регионе может способствовать или препятствовать реализации выбранной стратегии. При анализе условий спроса надо учесть наличие:

— разборчивых и требовательных покупателей продукции в определенных сегментах регионального рынка;

— в регионе потребителей продукции данной отрасли, которые могут быть инициированы и в других регионах;

— в регионе развитых каналов распределения и сбыта.

Анализ сопутствующих и поддерживающих отраслей должен дать ответы на вопросы:

— имеет ли (или располагает устойчивыми связями) данный регион снабжающие отрасли соответствующего уровня и для каких сегментов отрасли;

— имеет ли регион сильные позиции в важных сопутствующих и поддерживающих отраслях?

Потенциальные возможности диверсификации деятельности предприятий предпринимательского кластера в регионе были исследованы автором на примере предприятий машиностроительного комплекса Пензенской области как наиболее развитой отрасли промышленности «Машиностроение и металлообработка» (доля в структуре промышленности региона составляет 36,5%). При этом исходная позиция анализа — это те виды деятельности, которые предприятия официально (открыто) заявляют как доступные для осуществления.

При анализе направлений диверсификации предприятий отраслевого регионального рынка применялся следующий методический подход. На первом этапе выбирались предприятия, относящиеся к отраслевому региональному рынку. На втором этапе анализировались направления потенциальной диверсификации по отрас-





лям с позиций связанности видов деятельности с основными видами. На следующем этапе была составлена «Карта предпочтений», в которой приведен предпочитаемый предприятиями перечень отраслей диверсификации и выполнен соответствующий количественный анализ.

В соответствии с предложенной методикой был проведен анализ по двадцати четырем предприятиям машиностроительного комплекса Пензенской области, который показал, что у 47,7 % анализируемых предприятий определены направления диверсификации, связанные с основным видом деятельности; 52,3 % предприятий диверсифицированы в несвязанные отрасли, что свидетельствует о вполне зрелом подходе к управлению бизнесом. В группе выделенных предприятий есть такие, которые имеют только один основной вид деятельности. Кроме того, существуют предприятия, которые диверсифицированы как в машиностроение, так и в другие отрасли.

Из всех предприятий машиностроительного комплекса 14 % относятся к не диверсифицированным, 5 % — к слабо диверсифицированным, 33 % имеют средний, 48 % — высокий уровень диверсификации (шесть и более видов деятельности).

Анализ показывает, что более 80 % предприятий стремятся расширить сферы своего бизнеса, выйти на новые рынки, получать дополнительную прибыль (развитие торговли, общественного питания, бытового обслуживания). Это можно расценивать как позитивный фактор.

«Карта предпочтений» диверсификации машиностроительного комплекса показала, что 52,4 % анализируемых предприятий готовы заниматься внешнеторговой деятельностью, что говорит о потенциале повышения конкурентоспособности выпускаемой ими продукции; 71,4 % предприятий занимаются розничной торговлей, что почти закономерно для подобных организаций, работающих в современных условиях. Достаточно большой процент занимает такой вид деятельности, как оптовая торговля, — 46,7 %. Интересно, что сравнительно незначительная часть (9,5 %) предприятий развивают маркетинговую деятельность как вид услуг, консультации по вопросам коммерческой деятельности, финансов, управления.

При проведении анализа 1125 предприятий промышленного комплекса Пензенской области было вы-

делено 161 предприятие, которые диверсифицированы в машиностроение, т. е. наблюдалась обратная диверсификация; 24,2 % предприятий промышленности строительных материалов и 12,5 % предприятий легкой промышленности диверсифицированы в машиностроительную отрасль. Это можно объяснить тем, что продукция машиностроения часто применяется строительными организациями, поэтому последние стремятся максимально снизить затраты и сами для себя изготавливают строительную оснастку.

Изучение потенциала любого кластера может позволить разработать бизнес-портфель отраслевого комплекса, при наложении которого на информацию о потребностях экономических агентов региона можно выявить первоочередные возможности более полного использования имеющегося потенциала, а также потенциальные направления диверсификации.

Основой достижения конкурентного преимущества является координация деятельности как отдельного предприятия, так и предприятий отраслевого кластера при осуществлении стратегии диверсификации, а следовательно, требуется создание организации поддержки диверсификации в регионе. С одной стороны, как объект с внутренней упорядоченной структурой, находящийся в единстве с внешней средой, а с другой — как социальный субъект, имеющий свою логику функционирования и развития в соответствии с заданной целевой функцией, предлагаемая организация может рассматриваться как социально-экономическая система.

Эффективность координационных усилий в процессе диверсификации на региональном рынке будет достигаться, когда предприятия отрасли будут соизмерять возможности собственного потенциала и потребности заинтересованных субъектов. При этом органы исполнительной власти могут прилагать усилия к изучению спроса в регионе, исходя из перспектив его развития, и частично решать организационные вопросы с представителями сопутствующих и поддерживающих отраслей, а также институциональных структур региона, а последние — выберут стратегию своего участия в поддержке диверсификации деятельности предприятий реального сектора в данном регионе.

☎ (841-2) 36-82-13

E-mail: [em@diamond.stup.ac.ru](mailto:em@diamond.stup.ac.ru)



**РОССИЙСКАЯ АКАДЕМИЯ НАУК**  
ИНСТИТУТ ПРОБЛЕМ УПРАВЛЕНИЯ им. В.А. ТРАПЕЗНИКОВА  
**МЕЖНАРОДНАЯ НАУЧНО-ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ**  
**"ТЕОРИЯ АКТИВНЫХ СИСТЕМ-2005"**

*Основные направления работы конференции:*

- ✓ базовые модели и механизмы теории активных систем;
- ✓ принятие решений и экспертные оценки;
- ✓ управление безопасностью;
- ✓ управление проектами;
- ✓ прикладные задачи теории активных систем.

*Время и место проведения:*

16–18 ноября 2005 г., Москва, Профсоюзная ул., 65, Институт проблем управления РАН

Более подробную информацию о конференции можно найти на сайте теории управления организационными системами: [www.mtas.ru](http://www.mtas.ru)

# УПРАВЛЕНИЕ ВЫБОРОМ ПОСТАВЩИКА В УСЛОВИЯХ ВЕРОЯТНОСТНОГО ПРОГНОЗА

Е. И. Макаров, П. А. Головинский

*Воронежский государственный архитектурно-строительный университет*

Рассмотрена модель вероятностного прогноза выбора наилучшего поставщика, в соответствии с которой динамика параметров поставщика описывается временными рядами, моделируемыми трехслойными искусственными нейронными сетями; поставщик оценивается путем вычисления функции полезности; при выборе поставщика рассчитывается вероятность его преимущества с учетом прогноза. Дан пример выбора наилучшего поставщика щебня в дорожном строительстве.

## ВВЕДЕНИЕ

Становление рыночной экономики в корне изменило характер взаимоотношений в строительном комплексе, и в первую очередь — в области управления материальными потоками строящихся объектов. Состояние спроса со стороны подрядных организаций и ценовая политика поставщиков стали основными факторами конъюнктуры рынка строительных материалов. В условиях экономики рыночного типа главная проблема для поставщиков состоит в управлении сбытом продукции, а для потребителей — в максимизации полезности ее приобретения.

При поиске эффективных управленческих решений в области материально-технического обеспечения строительства важно построение рациональных логистических систем, состоящих из функциональных элементов со своими многоуровневыми иерархическими структурами, характеризующимися функциональной и структурной стабильностью, в зависимости от выбранной стратегии, поставленной цели и временного интервала.

Одно из фундаментальных свойств логистической системы заключается в устойчивости, которую можно определить как постоянство, неизменность определенного состояния системы (статическая устойчивость) или перехода из любых других состояний в данное состояние (динамическая устойчивость).

Применение структурно-функциональных конструкций для анализа устойчивости функционирования логистических систем показало, что устойчивость определяется набором структурных и функциональных элементов, наличием между ними определенной взаимосвязи и вероятностью того, что значение принятого критерия оптимальности в данном состоянии системы выше, чем в каком-либо другом. При понижении установленного уровня система переходит в другое со-

стояние, характеризующееся сменой функциональных элементов.

В данной работе мы рассмотрим процесс перехода логистической системы в новое состояние в результате оптимального выбора поставщика по критерию максимума полезности в условиях вероятностного прогноза. Конкретная задача, которая будет рассмотрена в виде примера, — задача о выборе наилучшего поставщика щебня для строительства автомобильной дороги с учетом прогноза на месяц вперед.

## 1. ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛОГИСТИКИ

Нейронные сети позволяют эффективно моделировать нелинейные системы при скрытых зависимостях. Их применение дает возможность почти автоматически учесть различные нелинейные взаимосвязи между показателями — признаками, характеризующими данные [1]. Нейросеть позволяет реализовать довольно сложные математические модели с помощью программных средств нейропакетов [2–4]. Она строится на основе задания множества примеров, от качества подбора которых зависят качество и скорость обучения системы. Нейросети представляют собой альтернативу нелинейным моделям множественной и логистической регрессии, а также дискриминантному анализу (классификации). К достоинствам нейросетевого моделирования следует отнести возможность решения комплекса проблем в единой парадигме, а к недостаткам — затруднения с интерпретацией правил вывода и представлением этих правил в явном виде. Тем самым нейросетевые и статистические подходы к анализу и моделированию данных являются как конкурирующими, так и взаимодополняемыми.

Нейросетевой подход позволяет в более сжатые сроки решать сложные плохо формализованные задачи, включая построение экспертных систем. Методы раз-



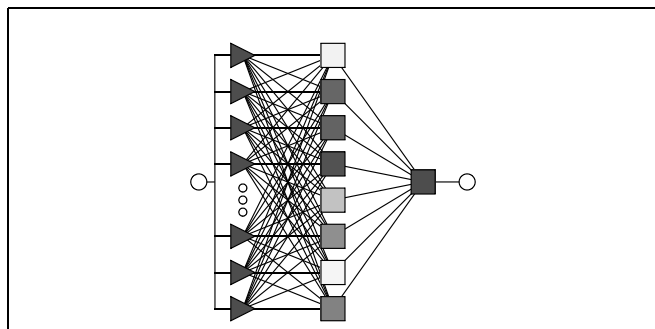
вернутого статистического анализа обеспечивают детальный анализ структуры данных и позволяют подбирать полуаналитические модели процессов. Такие полуаналитические модели дают возможность ответить на ряд важных практических вопросов. В решаемой нами задаче таким вопросом является устойчивость выбора поставщика и возможность изменения наиболее эффективного поставщика в результате динамического изменения параметров. Применим нейросетевой подход к анализу временных рядов.

## 2. МОДЕЛИРОВАНИЕ ВРЕМЕННОГО РЯДА С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

Задача моделирования временных рядов является частным случаем задачи регрессии и может быть решена с помощью нейронной сети любого типа, предназначенной для решения задач регрессии. Мы моделировали нейронные сети с помощью пакета STATISTICA: Neural Networks [5], который позволяет строить прогноз для нескольких выходных переменных, причем заданных не только в числовой, но и в номинальной форме. Можно строить прогноз как на один, так и на несколько шагов вперед. Для анализа временного ряда для одной переменной создается файл, соответствующий этой переменной. Для нейронной сети эта переменная служит одновременно и входной, и выходной (разумеется, в разные моменты времени).

В задаче прогноза временного ряда необходимо задать число значений, по которым будет строиться прогноз, а также число значений вперед для определения дальности прогноза. Поскольку в наших данных присутствует сезонная составляющая, то задается временное окно и горизонт прогноза на месяц вперед. Для моделирования временного ряда применялось последовательное построение множества сетей, из которых выбиралась наиболее эффективная по точности. Такой сетью оказалась трехслойная нейронная сеть на основе персептронов (Multilayer Perceptron).

Для обучения сети задавалось число обучающих (не больше длины ряда) и контрольных (остаточных) членов ряда наблюдений. Далее сеть обучалась либо методом сопряженных градиентов, либо методом Левенберга—



Структура нейронной сети, моделирующей временной ряд:

1 — треугольниками в левом входном слое обозначены входные нейроны; 2 — квадраты в среднем слое обозначают скрытые нейроны; 3 — выходной слой справа состоит из одного нейрона

Таблица 1

Значения прогнозируемых параметров

Параметр	Поставщик	Значение параметра	Значение дисперсии
Отпускные цены	1	253,41	1,45
	2	271,92	3,79
	3	237,97	2,38
	4	260,88	2,68
Предоплата	1	100	0
	2	0	0
	3	49,43	0,825
	4	0	0
Качество по зерновому составу	1	10,053	0,21
	2	14,27	0,34
	3	0	0
	4	0	0
Качество по прочности	1	1400	0
	2	1200	0
	3	1000	0
	4	1200	0
Надежность поставок	1	90,42	1,52
	2	85,34	1,34
	3	77,21	1,41
	4	74,42	1,57
Рентабельность предприятия	1	10,31	2,51
	2	12,71	3,02
	3	15,63	2,74
	4	9,4	1,89
Ликвидность	1	0,26	0,051
	2	0,20	0,033
	3	0,18	0,025
	4	0,14	0,015

Маркара. Наилучшие результаты по скорости обучения и сходимости процесса показал метод сопряженных градиентов. Полученная ошибка аппроксимации ряда служит критерием для выбора наилучшей сети и применена далее для анализа устойчивости выбора наилучшего поставщика.

Структура наилучшей нейронной сети, полученной при моделировании временных рядов показана на рисунке.

В табл. 1 приведены результаты прогноза на месяц вперед семи параметров, характеризующих поставки щебня с четырех различных предприятий: 1 — ОАО «Павловскгранит»; 2 — ОАО «Руда»; 3 — Новолипецкий металлургический комбинат; 4 — Лебединский ГОК.

### 3. УСТОЙЧИВОСТЬ ВЫБОРА НАИЛУЧШЕГО ПОСТАВЩИКА

Для оценки качества поставщика мы пользовались функциями частичной полезности, каждая из которых характеризует качество поставщика по одному из оценочных параметров. Функция полной полезности  $F$  в принятой нами аддитивной модели представляет собой сумму функций частичной полезности  $Q_i$ , взятых с весовыми коэффициентами  $a_i$ , определяющих меру влияния факторов, т. е.

$$F = \sum_i a_i Q_i(x_i), \quad (1)$$

где  $i$  — номер частичной полезности,  $i = 1, \dots, n$ ;  $x_i$  — аргумент (параметр), от которого зависит функция частичной полезности. Будем нумеровать поставщиков верхним индексом  $j = 1, \dots, N$ , где  $N$  число поставщиков. Тогда в данный момент времени для  $j$ -го поставщика полная функция полезности

$$F^j = \sum_i a_i Q_i(x_i^j). \quad (2)$$

Задача определения наилучшего поставщика соответствует нахождению номера  $j$ , при котором выражение (2) достигает своего наибольшего значения

$$F^k = \max_j F^j. \quad (3)$$

В результате временных изменений каждый параметр  $x_i^j$  можно рассматривать как временной ряд

$$x_i^j = x_i^j(t). \quad (4)$$

Временной ряд представляется нами с помощью нейронных сетей, как это описано выше. Наличие такой модели временного ряда позволяет предсказать выбор наилучшего поставщика в последующие моменты времени. Для этого достаточно подставить значения аргументов (4) из модели временного ряда, соответствующие данному моменту времени, в функции (1), (2) и в соответствии с условием (3) определить наилучшего поставщика.

Если учитывать только детерминированную составляющую временного ряда, то прогноз наилучшего поставщика и его возможной смены будет носить также детерминированный характер. В действительности временные ряды, описывающие динамику параметров задачи, являются случайными функциями времени. Поэтому более адекватный прогноз их динамики будет содержать случайную компоненту.

Пусть  $\delta$  — случайная компонента временного ряда, задающая неопределенность прогноза. Тогда погрешность детерминированной модели в оценке функции полезности тогда можно записать в виде

$$\delta F^j = \sum_i [F^j(x_i^j + \delta_i^j) - F^j(x_i^j)].$$

Приближенно это выражение можно представить как полный дифференциал функции многих переменных [6]:

$$\delta F^j \approx \sum_i \frac{\partial F^j(x_i^j)}{\partial x_i^j} \delta_i^j = \sum_i a_i \frac{\partial Q^j(x_i^j)}{\partial x_i^j} \delta_i^j. \quad (6)$$

Тем самым каждая оценка значения функции полезности будет представлена теперь не точкой, а интервалом  $(\delta F^j(x_i^j) - \delta_i^j, \delta F^j(x_i^j) + \delta_i^j)$ . Такая процедура линеаризации задачи в окрестности некоторых значений параметров широко применяется в теории вероятностей [7]. Если различные параметры  $x_i^j$  статистически независимы для разных индексов  $i$ , то оценка

$$\delta F^j = \sqrt{\sum_i a_i^2 \left( \frac{\partial Q^j(x_i^j)}{\partial x_i^j} \delta_i^j \right)^2},$$

как это имеет место в обычной теории погрешностей.

Поскольку нас интересует наилучший поставщик в данный момент времени, то теперь для его определения нам необходимо проанализировать структуру интервалов значений функции полезности для всей совокупности поставщиков. Наилучшего поставщика можно выбрать путем парного сравнения поставщиков.

Рассмотрим задачу для пары поставщиков, которых мы обозначим индексами 1 и 2. Пусть эти поставщики характеризуются значениями случайных функций полезности  $F^1$  и  $F^2$ . Обозначим средние значения этих величин  $\langle F^1 \rangle$  и  $\langle F^2 \rangle$ , а их дисперсии  $\delta F^1$  и  $\delta F^2$ . Образующиеся интервалы погрешностей могут перекрываться различным образом. На самом деле значения  $F^1$  и  $F^2$  могут отклоняться от средних значений этих величин случайным образом с некоторой вероятностью, которая уменьшается по мере увеличения отклонения. Нам необходимо определить полную вероятность того, что  $F^1 < F^2$  в некоторый начальный момент времени  $t = 0$ . Это означает смену наилучшего поставщика 1 на нового лучшего поставщика 2 с некоторой вероятностью. Если вероятность меньше 0,5, то смена поставщика явно нецелесообразна. Для выбора критерия смены поставщика необходимо определить практически рациональный критерий значения вероятности, который разумно принять на уровне около 0,9.

Определим вероятность смены поставщика, исходя из предположения, что  $F^1$  и  $F^2$  — независимые случайные функции с нормальным распределением. Обозначим плотность вероятности распределения величины  $F^1$  в момент времени  $t$  как  $\rho_1(F^1)$ , а плотность вероятности распределения величины  $F^2$  в тот же момент времени как  $\rho_2(F^2)$ . Вероятность того, что величина  $F^2$  принимает некоторое значение в бесконечно малом интервале  $(F^2, F^2 + \Delta F^2)$  равна  $\Delta W^2 = \rho_2(F^2) \Delta F^2$ . Вероятность того, что при этом  $F^1 < F^2$ , равна интегралу

$$W^1 = \int_{-\infty}^{F^2} \rho_1(F^1) dF^1.$$

Вероятность наступления двух таких событий одновременно есть  $\Delta W = W^1 \Delta W^2$ .

Полная вероятность наступления события  $F^1 < F^2$ , т. е. смены поставщика, равна интегралу

$$W = \int_{-\infty}^{\infty} \rho_2(F^2) \left( \int_{-\infty}^{F^2} \rho_1(F^1) dF^1 \right) dF^2.$$





Полученный результат можно выразить через функции распределения [8]:

$$\Phi_i(F^i) = \int_{-\infty}^{F_i} \rho_i(F) dF. \quad (7)$$

С учетом определения (7) получим

$$W = \int_{-\infty}^{\infty} \Phi_1(F^2) d\Phi_2(F^2).$$

Пусть теперь плотности вероятности — гауссовские функции, т. е. соответствуют нормальному закону распределения. В этом случае

$$\rho_i(F_i) = \frac{1}{\delta F_i \sqrt{2\pi}} \exp \left[ -\frac{1}{2} \left( \frac{F^i - \langle F_i \rangle}{\delta F^i} \right)^2 \right].$$

Соответственно,

$$\Phi_i(F^i) = \int_{-\infty}^{F^i} \rho_i(F) dF = \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{F^i - \langle F_i \rangle}{\delta F^i} \right) \right],$$

где  $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-u^2) du$  — функция ошибок, табули-

рованная и задаваемая интегральным представлением [8]. Она принимает значения в интервале от 0 при  $z = 0$  до 1 при  $z = \infty$ . Соответственно, вероятность смены поставщика

$$W = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-y^2/2) \times \left[ 1 + \operatorname{erf} \left( y \frac{\delta F^1}{\delta F^2} + \frac{\langle F^1 \rangle - \langle F^2 \rangle}{\delta F^2} \right) \right] dy. \quad (8)$$

Очевидно, что при  $(\langle F^1 \rangle - \langle F^2 \rangle) / \delta F^2 \gg \delta F^2 / \delta F^1$  вероятность смены поставщика достигает значения  $W = 1$ .

Если рассчитать таким же способом вероятность того, что наилучшим поставщиком будет (останется) первый поставщик, т. е.  $F^2 \leq F^1$ , то вероятность такого события

$$W' = \int_{-\infty}^{\infty} \rho_2(F^2) \left( \int_{F^2}^{\infty} \rho_1(F^1) dF^1 \right) dF^2.$$

Легко убедиться, что  $W + W' = 1$ , т. е., что события  $F^1 < F^2$  и  $F^1 \geq F^2$  образуют полную систему событий, и наступление хотя бы одного из них есть достоверное событие с вероятностью 1.

Для расчета прогноза выбора наилучшего поставщика на месяц вперед по результатам прогноза характерис-

тических параметров, приведенных в табл. 1, необходимо вычислить как значения функции полной полезности, так и определить вероятность предпочтения того или иного поставщика. Весовые коэффициенты для вычисления функции полезности по функциям частичной полезности определялись экспертным путем и для данного примера приведены в табл. 2.

Вычисление вероятности  $W$  связано с численным интегрированием выражения (8), содержащего специальную функцию ошибок и осуществлялось нами в пакете MAPLE 7.0. Значения функций частичной полезности и их производных приведены в Приложении.

Вычисленные значения функции полезности в заданном порядке составляют  $F^1 = 0,441$ ;  $\delta F^1 = 0,0127$ ;  $F^2 = 0,381$ ;  $\delta F^2 = 0,011$ ;  $F^3 = 0,440$ ;  $\delta F^3 = 0,009$ ;  $F^4 = 0,526$ ;  $\delta F^4 = 0,006$ . Вероятность преимущества первого поставщика по сравнению со вторым составляет 0,999 и он обладает очевидным преимуществом, так же, как и третий поставщик в сравнении со вторым. Вероятность преимущества первого поставщика по сравнению с третьим составляет 0,531, т. е. они практически эквивалентны. Вероятность преимущества четвертого поставщика по сравнению с третьим составляет единицу. Таким образом, устойчивым наилучшим в рассмотренном нами примере является четвертый поставщик — Лебединский ГОК.

## ЗАКЛЮЧЕНИЕ

Построенная нами модель прогноза и анализа параметров поставщиков позволяет давать оценку наилучшего поставщика на основе оценки вероятности. Качество прогноза в значительной степени определяется качеством модели временного ряда. Моделирование на основе нейронных сетей позволяет построить совокупность таких сетей и отобрать из них наилучшую, обеспечивающую представление ряда и его прогноз с наибольшей точностью. Вычисление вероятности при выборе наилучшего поставщика основывалось на предположении о непрерывности всех величин в задаче, а также на предположении о непрерывности производных функций частичных полезностей. В действительности уже в нашей модели функции рентабельности производства и коэффициент абсолютной ликвидности являются кусочно-непрерывными, а их производные терпят разрывы. Тем самым, принятая нами модель в окрестностях точек разрыва может приводить к большим погрешностям при оценке поставщиков. Построение модели прогноза с учетом возможности резких, катастрофических изменений в структуре временных рядов и поведении функции полезности представляет самостоятельную задачу дальнейших исследований.

## ПРИЛОЖЕНИЕ

Поскольку для вычисления оценки дисперсии функции полезности требуются значения производных функций частичной полезности, приведем выражения для функций и их производных.

Таблица 2

Значения весовых коэффициентов

Номер коэффициента	1	2	3	4	5	6	7
Его значение	0,3	0,1	0,1	0,2	0,1	0,1	0,1

Критерий	Функция полезности	Ее производная	Примечания
Отпускная цена	$(x - b)^2 / (a - b)^2, x \in [a, b]$	$2 \frac{x - b}{(a - b)^2}$	Значения $a$ и $b$ выбираются как минимальное и максимальное по всем поставщикам за весь период времени
Предоплата	$1 - (x/b)^2, x \in [0, b]$	$-2x/b^2$	$b = 100$
Качество по зерновому составу	$(a - x)/(b - a), x \in [a, b]$	$\frac{1}{a - b}$	Значения $a$ и $b$ выбираются как минимальное и максимальное по всем поставщикам за весь период времени
Качество по прочности	$(x - a)/(b - a), x \in [a, b]$	$\frac{1}{b - a}$	То же
Надежность поставок	$1 - (x/100 - 1)^2, x \in [0, 100]$	$2(1 - x/100)/100$	—
Рентабельность предприятия	$\begin{cases} 0, & x \leq 0 \\ x/a, & 0 < x < a \\ 1, & x \geq a \end{cases}$	$\begin{cases} 0, & x \leq 0 \\ 1/a, & 0 < x < a \\ 0, & x \geq 0 \end{cases}$	Выбирается среднеотраслевое значение $a = 35$
Ликвидация	$\begin{cases} x/a, & 0 \leq x < a \\ 1, & x \geq a \end{cases}$	$\begin{cases} 1/a, & 0 \leq x < a \\ 0, & x \geq a \end{cases}$	Значение $a = 0,5$ нормативный максимум

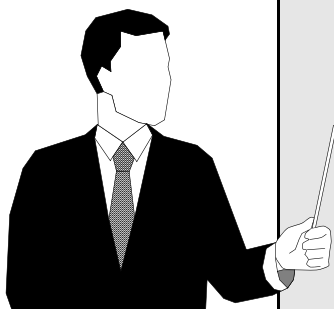
## ЛИТЕРАТУРА

1. Степанов В. С. Фондовый рынок и нейросети // Мир ПК. — 1998. — № 12. — С. 1—9.
2. Круглов В. В., Борисов В. В. Искусственные нейронные сети. — М.: Горячая линия — Телеком, 2001. — 382 с.
3. Осовский С. Нейронные сети для обработки информации. — М.: Финансы и статистика, 2002. — 344 с.
4. Медведев В. С., Потемкин В. Г. Нейронные сети. Матлаб 6. — М.: Диалог МИФИ, 2002. — 496 с.
5. Нейронные сети. STATISTICA: Neural Networks. — М.: Горячая линия — Телеком, 2001. — 182 с.
6. Смирнов В. И. Курс высшей математики. Т. 1. — М.: ГИТТЛ, 1951. — 472 с.
7. Венцель Е. С. Теория вероятностей. — М.: Высшая школа, 1998. — 576 с.
8. Корн Г., Корн Т. Справочник по математике. — М.: Наука, 1968. — 720 с.

☎ (0732) 52-47-34

E-mail: [golovinski@mail15.com](mailto:golovinski@mail15.com);

[golovinskip@mail.ru](mailto:golovinskip@mail.ru) □



### Читайте в следующем номере

**К 100-летию академика В.А. Трапезникова**

**Эпштейн В.Л.** Электронная гиперкнига - новая эпоха в истории науки и обучения

**Антонова Г.М., Цвиркун А.Д.** Оптимизационно-имитационное моделирование для решения проблем оптимизации современных сложных производственных систем

**Прангишвили И.В.** Повышение эффективности управления сложными организационными и социально-экономическими системами

**Пантелеев Е.А.** Роль государства в формировании благоприятной институциональной среды, обеспечивающей развитие промышленного комплекса региона (на примере г. Москвы)

**Коврига С.В.** Методические и аналитические основы когнитивного подхода к SWOT-анализу

**Иващенко А.А., Нижегородцев Р.М., Новиков Д.А.** Инновационная и инвестиционная политика: модель смены технологий

# УПРАВЛЕНИЕ ПОДВИЖНЫМИ ОБЪЕКТАМИ В УСЛОВИЯХ ИСКУССТВЕННО ОРГАНИЗОВАННОЙ НЕПОЛНОТЫ ИНФОРМАЦИИ<sup>1</sup>

Т. Г. Абрамянц, Е. П. Маслов, Е. Я. Рубинович

*Институт проблем управления им. В. А. Трапезникова, г. Москва*

Рассмотрены задачи конфликтного управления подвижными объектами в условиях радиоэлектронного противодействия. Поведение противников моделируется дифференциальными играми с групповой целью и дифференциальными играми в смешанных стратегиях.

## ВВЕДЕНИЕ

Задачи управления в условиях искусственно организованной неполноты информации являются математическими моделями конфликтного управления в условиях радиоэлектронного противодействия (РЭП).

Чрезвычайная актуальность задач РЭП привела к тому, что к их решению были привлечены специалисты различных научно-технических дисциплин, в том числе, и теории управления.

Существуют различные способы воздействия средств РЭП на информационные и управляющие системы противника [1]. Основные из них заключаются в использовании имитирующих и подавляющих воздействий. Имитирующее воздействие сводится к созданию системы подвижных или неподвижных объектов материальной и(или) волновой природы, воспроизводящих с той или иной степенью подобия основные свойства защищаемого объекта. Радиоэлектронное подавление осуществляется путем создания помех в информационных каналах противника.

По инициативе А. М. Петровского, в Институте проблем управления возникло и развивается направление исследований, связанных с оптимизацией законов управления подвижными объектами в условиях РЭП.

Для оптимизации законов управления в условиях имитирующего воздействия был разработан новый раздел теории дифференциальных игр преследования-ук-

лонения — дифференциальные игры с групповой целью. Для оптимизации законов управления в условиях подавляющего воздействия был привлечен математический аппарат дифференциальных игр в смешанных стратегиях.

Настоящая работа содержит обзор основных результатов, полученных в ходе исследований.

## 1. ДИФФЕРЕНЦИАЛЬНЫЕ ИГРЫ С ГРУППОВОЙ ЦЕЛЬЮ

Дифференциальные игры с групповой целью являются математическими моделями задач преследования—уклонения, в которых уклоняющаяся истинная цель использует в ходе преследования ложные цели. Существенный момент в таких играх состоит в том, что число преследователей меньше числа целей. Наличие ложных целей приводит к информационной дискриминированности преследующей стороны, поскольку ей с достоверностью не известно, какая же из целей истинная. В связи с этим преследующая сторона должна затратить определенные ресурсы для обнаружения, классификации и последующего догона истинной цели или отказаться от попыток классификации целей и сблизиться (в некоторой метрике) с коалицией уклоняющихся в целом.

В задачах преследования-уклонения с групповой целью имеют смысл естественные для игр преследования—уклонения критерии типа “промах” или “время преследования”, которые порождают, соответственно, два класса задач:

— дифференциальные игры совместного преследования [2—6];

— дифференциальные игры поочередного преследования [2, 3, 6—10].

<sup>1</sup> Работа доложена на Научных чтениях памяти профессора А. М. Петровского, Москва, Ин-т проблем управления, 17 марта 2005 г.

### 1.1. Дифференциальные игры совместного преследования

Множество дифференциальных игр совместного преследования можно условно разбить на два подкласса:

- однокритериальные игры [2—6, 11],
- двухкритериальные игры [6, 12, 13].

**Однокритериальные игры.** Плата в однокритериальных играх совместного преследования формируется как взвешенная сумма терминальных (конечных) расстояний от преследователя до  $i$ -й цели:

$$G = \sum_{i=1}^N p_i |z_i(T)|; \quad \sum_{i=1}^N p_i = 1. \quad (1)$$

Здесь и далее  $T$  — момент окончания игры,  $z_i(T)$ ,  $i = 1, \dots, N$  — вектор, направленный от преследователя к  $i$ -й цели. Весовые коэффициенты  $p_i \geq 0$  имеют смысл вероятности того, что  $i$ -я цель — истинная. Физический смысл критерия — математическое ожидание расстояния от преследователя до истинной цели в момент окончания игры. Плату  $G$  преследователь минимизирует, коалиция целей — согласованно максимизирует.

Идея организации процесса совместного преследования состоит в том, что в ходе его происходит сближение преследователя с группой целей в целом — а значит, происходит сближение и с истинной целью.

Введение в плату  $G$  вероятностей  $p_i \geq 0$  позволяет учитывать информационные возможности преследователя по части классификации целей. Эти вероятности могут оставаться неизменными или меняться в ходе преследования, например, могут зависеть от текущих расстояний до целей [2].

Основная особенность однокритериальных игр совместного преследования заключается в том, что моментом окончания игры (прекращением преследования) распоряжается преследователь.

В зависимости от характера имеющейся информации управление преследователя может быть программным (зависящим лишь от начальных позиций противников) или позиционным (с обратной связью по текущим положениям противников). В первом случае преследователь выбирает закон управления движением и длительность игры в начальный момент времени, во втором — имея информацию о текущих позициях сторон, останавливает игру по достижении определенного взаимного положения игроков (т. е. при выходе состояния игры на некоторую терминальную поверхность).

В классе однокритериальных игр совместного преследования больше всего результатов получено для случая противодействия одного преследователя и группы из двух целей. Здесь решены задачи с фиксированной и нефиксированной заранее (выбираемой преследователем) длительностью игры [2—4]. Найдены решения при программных и позиционных законах управления траекторией преследования, при безынерционных (“простых”) — по терминологии работы [14]) движениях игроков и при ограничениях на кривизну их траекторий.

Приведем в качестве примера решение плоскостной дифференциальной игры совместного преследования в позиционных стратегиях при “простых” движениях иг-

роков и постоянных вероятностях  $p_i$ . Динамика игры описывается соотношениями

$$\dot{z}_i = v_i - u, \quad z_i(0) = z_i^0, \quad i = 1, 2, \quad (2)$$

где  $v_i$ ,  $u$  — двумерные векторы управлений (скоростей) игроков:  $i$ -й цели  $E_i$  и преследователя  $P$ , соответственно, стесненные ограничениями

$$|u| \leq 1; \quad |v_i| \leq \beta_i \leq 1, \quad (3)$$

$z_i^t \triangleq z_i(t)$  — двумерный вектор, направленный от преследователя  $P$  к цели  $E_i$  в текущий момент  $t$ . Здесь и далее верхний индекс указывает момент времени.

Решение задачи имеет следующий вид.

Оптимальными стратегиями игроков являются движения по прямым с максимальными скоростями.

При

$$\beta = p_1 \beta_1 + p_2 \beta_2 \geq |p_2 - p_1|, \quad (4)$$

в зависимости от начального расположения целей, игра заканчивается точечной встречей с одной из целей или в момент выхода фазового состояния игры на терминальную поверхность

$$\frac{\langle z_1^T, z_2^T \rangle}{|z_1^T| \cdot |z_2^T|} = \frac{\beta^2 - p_1^2 - p_2^2}{2p_1 p_2} \quad (5)$$

где  $\langle \cdot, \cdot \rangle$  — символ скалярного произведения векторов,  $z_i^T = z_i(T)$ .

Решение задачи (1)—(5) иллюстрируется рис. 1. В неподвижной системе координат, связанной с начальным положением преследователя  $P^0$  (здесь и далее верхними индексами отмечены положения игроков в соответствующие моменты времени), изображено разбиение области начальных положений дальней цели  $E_2^0$ , т. е. внешности круга радиуса  $\rho_1 = |z_1^0|$ , на зоны  $A$ ,  $B$ ,  $D_1$  и  $D_2$  назначения оптимальных управлений игроков. Границами зон служат лучи  $L_1$ ,  $L_2$  и  $L_3$ , проведенные из точек

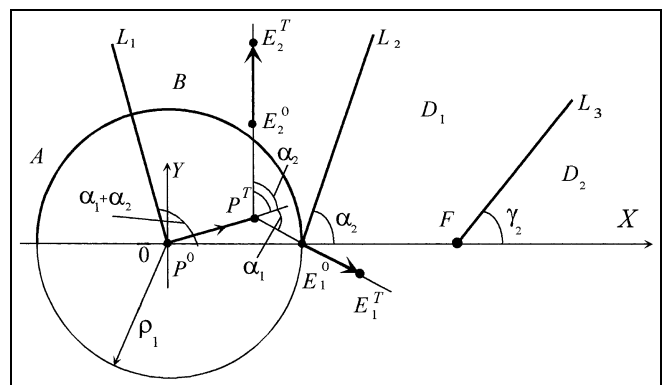


Рис. 1. Разбиение области начальных позиций целей на зоны оптимальных маневров





0,  $E_1^0, F = (\rho_1 + h, 0)$  под углами  $\alpha_1 + \alpha_2$ ,  $\alpha_2$  и  $\gamma_2$ , которые вычисляются в явном виде из соотношений

$$\cos \alpha_1 = (\beta^2 + p_1 - p_2) / (2p_1\beta),$$

$$\cos \alpha_2 = (\beta^2 + p_2 - p_1) / (2p_2\beta),$$

$$\sin \gamma_2 = (p_1 - \beta_1\beta) \sin(\alpha_1 + \alpha_2) / (\beta_1(1 + \beta_1^2 - 2\beta_1 \cos \alpha_1)^{1/2}),$$

где  $h = \rho_1\beta_1\beta / (p_1 - \beta_1\beta)$ . При  $E_2^0 \in A$  длительность игры равна нулю. При  $E_2^0 \in B$ ;  $E_2^0 \in D_1$  игра заканчивается реализацией терминального условия (5). При  $E_2^0 \in D_2$  преследователь осуществляет встречу с ближней целью.

В случае нарушения неравенства (4) определяющими являются не геометрические, а вероятностные характеристики целей. При любых начальных положениях целей преследователь осуществляет точечную встречу с той целью, вероятность  $p_i$  классификации которой как истинной выше.

**Двухкритериальные игры** совместного преследования являются математическими моделями задач уклонения подвижных объектов от обнаружения с применением средств РЭП. Необходимость в средствах РЭП возникает в тех случаях, когда начальное положение уклоняющегося объекта принадлежит *области захвата* преследователя. Такое название получила подвижная область в пространстве поиска, прилегающая к области обнаружения преследователя и обладающая тем свойством, что из точек этой области уклонение невозможно [3, 13]. Структура зоны захвата зависит от формы зоны обнаружения, динамических характеристик противодействующих объектов, вида траекторий поиска.

Далее приводятся постановка и решение двухкритериальной плоскостной дифференциальной игры [13].

Три объекта — преследователь  $P$ , истинная цель  $E_1$  и ложная цель  $E_2$ , обладая ограниченными скоростями, перемещаются по плоскости, имея возможность в каждый момент времени произвольно изменять направления своих движений.

Преследователь снабжен круговой зоной обнаружения радиуса  $R$ , центр которой совпадает с его текущим положением. Цель обнаруживается при попадании ее на границу зоны обнаружения.

До момента обнаружения какой-либо из целей преследователь движется по прямой с максимальной скоростью. После обнаружения цели — по траектории, соответствующей минимуму времени до классификации обнаруженной цели. Цель классифицируется преследователем с расстояния  $r \leq R$ .

Вводится неподвижная прямоугольная система координат  $XOY$ , начало которой совмещено с исходным положением преследователя  $P^0$ , а положительное направление оси абсцисс совпадает с вектором скорости преследователя при  $t = 0$  (рис. 2).

Прямолинейной поисковой траектории преследователя соответствует зона захвата, имеющая вид симметричного криволинейного треугольника  $ABCD$ , ограниченного дугой окружности и двумя касательными к окружности, пересекающимися в точке с координатами  $(R/\beta_1, 0)$ , где  $\beta_1$  — отношение скоростей истинной цели и преследователя.

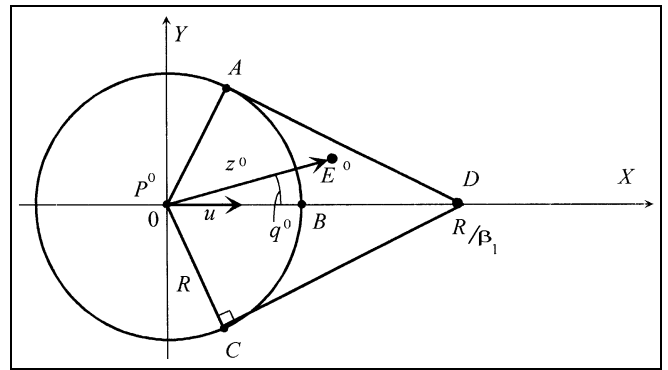


Рис. 2. Структура зоны захвата одиночной цели

В начальный момент времени обе цели находятся в одной точке  $E_1^0 = E_2^0 = E^0$  вне зоны обнаружения, но внутри области захвата (см. рис. 2).

Процесс противодействия преследователя и системы целей состоит из этапа поиска цели и этапа отвлечения (преследователя).

Динамика игры описывается соотношениями (2) и (3).

Коалиция целей согласованно максимизирует минимальное по времени расстояние между преследователем и истинной целью:

$$G_1 = \min_t |z_1(t)| \rightarrow \max_{v_1, v_2}$$

Преследователь минимизирует время до момента классификации обнаруженной первой (ложной) цели

$$G_2 = T - s \rightarrow \min_u$$

где моменты  $s$  и  $T$  определяются соотношениями  $|z_2(s)| = R$ ,  $|z_2(T)| = r$ .

Решение задачи имеет следующий вид.

Оптимальными стратегиями игроков являются движения с максимальными скоростями. Траектории преследователя и ложной цели состоят из двух отрезков прямых. Излом траекторий происходит в момент обнаружения преследователем ложной цели. Траектория истинной цели — прямолинейная. Минимальное расстояние между преследователем и истинной целью достигается на этапе отвлечения.

Система целей имеет два равноправных маневра, соответствующих выходу истинной цели из зоны захвата через одну или другую прямолинейную границу зоны. По оптимальной траектории истинная цель в течение всей игры движется перпендикулярно границе зоны захвата, построенной относительно траектории преследователя на этапе отвлечения.

На рис. 3 изображены кривые 1—3, соответствующие границам области начальных позиций внутри зоны захвата, из которой возможно уклонение истинной цели при наличии ложной цели. Кривые построены для  $\beta_1 = \beta_2 = 0,4$  и  $\sigma = r/R = 0$  (кривая 1),  $\sigma = 0,2$  (кривая 2),  $\sigma = 0,4$  (кривая 3). Уклонение возможно из точек, лежащих правее соответствующих кривых.

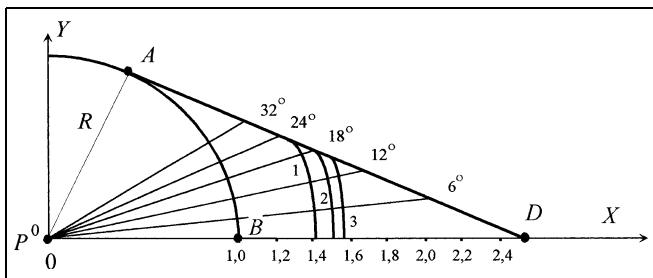


Рис. 3. Зоны захвата цели в условиях радиоэлектронного противодействия

Сокращение размера зоны захвата происходит вследствие того, что преследователь “отвлекается” на классификацию обнаруженной (ложной) цели.

Кривые на рис. 3 позволяют оценить степень сокращения области начальных позиций цели, из которых невозможно уклонение от обнаружения.

### 1.2. Дифференциальные игры поочередного преследования

В этих играх платой служит полное время поимки преследователем всех целей.

Специфика игр этого класса состоит в том, что в управление преследователя входит выбор очередности встреч с целями. Порядок встреч можно выбирать по-разному:

— в момент начала игры, т. е. программно [2, 7, 10, 11, 15–17];

— в процессе преследования как функцию текущих позиций игроков, т. е. позиционно [3, 9, 11].

В первом случае порядок встреч фиксируется в момент начала преследования и не меняется до конца игры. Во втором случае возможна смена порядка очередности встреч в ходе игры.

Далее приводятся постановка и решение плоскостных игр с программным и поочередным выбором порядка встреч при одинаковых максимальных скоростях целей:  $\beta_1 = \beta_2 = \beta$ .

Рассмотрим вначале игру поочередного преследования с программным выбором.

Три игрока — преследователь  $P$  и цели  $E_A, E_B$ , обладающая ограниченными скоростями, перемещаются по плоскости. Игроки обладают “простыми” движениями. Уравнения движения:

$$\dot{z}_i = v_i - u, \quad z_i(0) = z_i^0, \quad i = A, B, \quad (6)$$

$$|u| \leq 1, \quad |v_i| \leq \beta < 1. \quad (7)$$

Условия первой (в момент  $T_1$ ) и второй (в момент  $T_2$ ) встреч имеют вид:

$$z_j(T_1) = 0, \quad z_k(T_2) = 0, \quad (8)$$

где, в зависимости от очередности встреч,  $j = A, k = B$  или  $j = B, k = A$ .

Плата выражается функционалом

$$G = T_1 + T_{12} = T_2 \rightarrow \min \max_{u, k} v_A, v_B \quad (9)$$

где  $T_{12}$  — интервал времени между встречами.

Пусть в момент  $t = 0$  преследователь  $P$  принимает решение о том, что вначале происходит встреча с игроком  $E_A$ , а затем с игроком  $E_B$ .

Введем неподвижную систему координат  $XOY$ , начало которой совпадает с исходным положением преследователя  $P^0$ , а ось абсцисс проходит через начальное положение  $E_A^0$ . Пусть вторая цель  $E_B^0$  в начальный момент находится в верхней полуплоскости (рис. 4).

Решение игры (6)–(9) имеет следующий вид.

Оптимальными стратегиями игроков являются движения по прямым с максимальными скоростями. Геометрическим местом точек встречи преследователя и цели  $E_A$  является окружность Аполлония. Уравнение этой окружности

$$\left(x - \frac{|z_A^0|}{1 - \beta^2}\right)^2 + y^2 = |z_A^0|^2 \left(\frac{\beta}{1 - \beta^2}\right)^2.$$

Преследователь встречается с целью  $E_A$  в точке  $P^{T_1} = E_A^{T_1}$ . Цель  $E_B$  движется по прямой, соединяющей точки  $P^{T_1}$  и  $E_B^0$ . Справедливы равенства  $\nu = \mu = \psi_A$ , где  $\psi_A$  — угол между траекторией цели  $E_A$  и осью абсцисс (см. рис. 4),  $\nu, \mu$  — углы между радиусом окружности Аполлония, проведенным в точку  $P^{T_1}$ , и траекториями преследователя до и после встречи с целью  $E_A$  (см. рис. 4).

Цена игры

$$G_{AB}^* = \frac{|z_B^0| \sin(\delta^0 + \psi_A + \varphi)}{(1 - \beta) \sin \psi_A},$$

где  $\delta_0 = (\angle z_A^0, z_B^0)$ ,  $\varphi$  — угол между траекторией преследователя и осью абсцисс, углы  $\nu$  и  $\mu$  удовлетворяют условию оптимального “отражения” траектории преследователя от окружности Аполлония, звездочкой снабжены оптимальные величины.

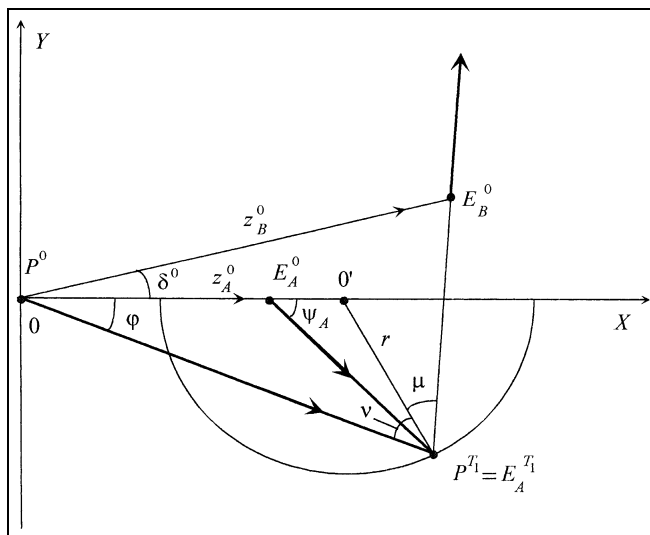


Рис. 4. Структура оптимальных траекторий при программном выборе очередности встреч с целями



Аналогично решается задача с платой  $G_{BA}$  (вначале преследователь встречается с целью  $E_B$ ). Оптимальная последовательность встреч определяется в момент  $t = 0$  из условия  $\min(G_{AB}^*, G_{BA}^*)$ , что в данном случае эквивалентно условию  $\min(|z_A^0|, |z_B^0|)$ : преследователь встречается вначале с тем игроком, до которого начальное расстояние меньше.

Обозначим этого игрока индексом 1, дальнего — индексом 2.

Перейдем к игре с позиционным выбором очередности.

В игре с программно выбранной очередностью встреч знак разности  $|z_2(t)| - |z_1(t)|$  в процессе движения игроков может, вообще говоря, изменяться. Позиционное управление по индексу  $k$  позволяет учесть тот факт, что при  $|z_2(t)| = |z_1(t)|$  в игре, начинающейся в момент  $t$ , цена как функция текущего состояния  $\{z_2(t), z_1(t)\}$  не зависит от очередности встреч. В смысле платы игроки  $E_1$  и  $E_2$  в этот момент идентичны для преследователя, и он может переиндексировать их, т. е. изменить порядок встреч.

Смена очередности не произойдет, если на оптимальных управлениях  $u^*, v_1^*$  и  $v_2^*$ , соответствующих программному выбору очередности встреч, в течение всего интервала  $[0, T_1]$  реализуется неравенство  $|z_2(t)| \geq |z_1(t)| \geq 0$ .

Обозначим символом  $\Gamma^0 = \{x_2^0, y_2^0\}$  множество начальных позиций дальней цели  $E_2^0$ , для которых на оптимальных управлениях, соответствующих программному выбору очередности, выполняется неравенство

$$|z_2(t)| - |z_1(t)| \geq 0, \quad t \in [0, T_1], \quad (10)$$

и, кроме того, существует момент

$$\theta = \inf\{t: |z_2(t)| - |z_1(t)| = 0\}, \quad t \in [0, T_1].$$

В статье [9] показано, что множество  $\Gamma^0$  имеет вид отрезка прямой, пересекающей ось абсцисс и касающейся окружности радиуса  $|z_1^0|$ , — см. отрезок  $M^0 N^0$  на рис. 5.

Угловая координата точки касания  $\delta^* = \pi/2 - \varphi^* - 2\psi_1^*$ , где углы  $\varphi^*$  и  $\psi_1^*$  однозначно определяются следующей системой уравнений:

$$\begin{aligned} \sin\varphi - \beta\sin\psi_1 &= 0, \\ \cos\varphi - \beta\cos\psi_1 &= \sin 2\psi_1, \quad \varphi, \psi_1 \geq 0. \end{aligned}$$

Отрезок  $M^0 N^0$  разбивает все множество начальных позиций дальней цели  $E_2^0$  (внешность круга радиуса  $|z_1^0|$  в верхней полуплоскости) на две зоны:  $C$  и  $D$ .

При  $E_2^0 \in D$  (внешность криволинейного треугольника  $E_1^0 M^0 N^0$ ) очередность встреч фиксируется в момент  $t = 0$  и в процессе всего преследования выполняется строгое неравенство (10).

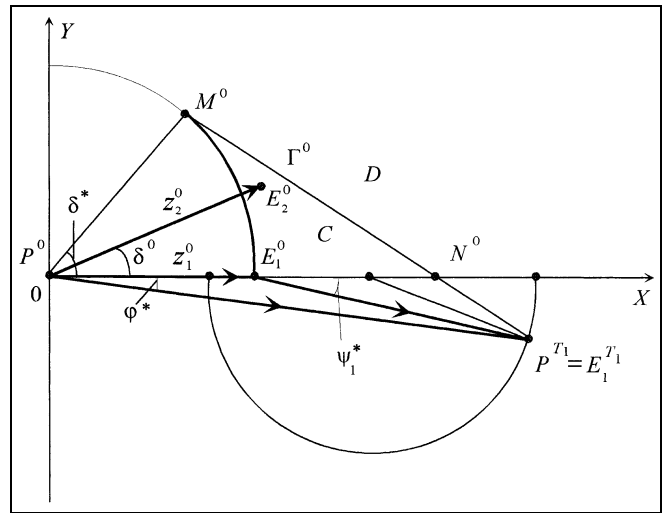


Рис. 5. Структура области начальных позиций целей при наличии этапа совместного преследования

При  $E_2^0 \in C$  (внутренность треугольника) в игре найдется момент  $\theta$  такой, что  $E_2^0 \in M^0 N^0 \triangle \Gamma^0$  и, кроме того,

$$|z_1^0| = |z_2^0|, \quad (z_1^0, z_2^0) = \delta^*,$$

где  $z_i^0 \triangle z_i(\theta), i = 1, 2$ .

Момент  $\theta$  разбивает игру на два этапа  $[0, \theta]$  и  $[\theta, T]$ .

Цена игры поочередного преследования на интервале  $[\theta, T]$   $G_\theta = \sigma|z_1^0|$ , где  $\sigma = (1 - \beta)^{-1} \text{ctg}\psi^*$ .

Полное время поочередного преследования в игре с позиционным выбором очередности  $G = \theta + \sigma|z_1^0|$ .

Игру на интервале  $[0, \theta]$  можно рассматривать как совместное преследование игроком  $P$  группы целей  $E_1$  и  $E_2$ . Этапу совместного преследования можно дать следующую интерпретацию: в течение всего интервала  $[0, \theta]$  преследователь  $P$  держит игроков  $E_1$  и  $E_2$  в условиях неопределенности об очередности встреч; окончательное решение о порядке встреч с целями принимается лишь в момент  $\theta$ . Созданием этой неопределенности преследователю удастся уменьшить цену игры.

Рис. 6 иллюстрирует структуру оптимальных траекторий игроков в игре поочередного преследования с позиционным выбором очередности для случая, когда в начальный момент обе цели находятся в одной точке  $E_{1,2}^0$ . На интервале времени  $[0, \theta]$  преследователь движется по прямой  $P^0 E_{1,2}^0$ . Цели расходятся симметрично относительно этой прямой,  $\alpha_1 = -\alpha_2$ . В момент  $\theta$ , находясь в позиции  $P^0$ , преследователь выбирает очередность встреч с целями. Рис. 6 соответствует случаю, когда вначале осуществляется точечная встреча с целью  $E_1$ . Заштрихованный треугольник  $\Delta M^0 E_1^0 N^0$  есть зона  $C$ , построенная в момент  $\theta$ .

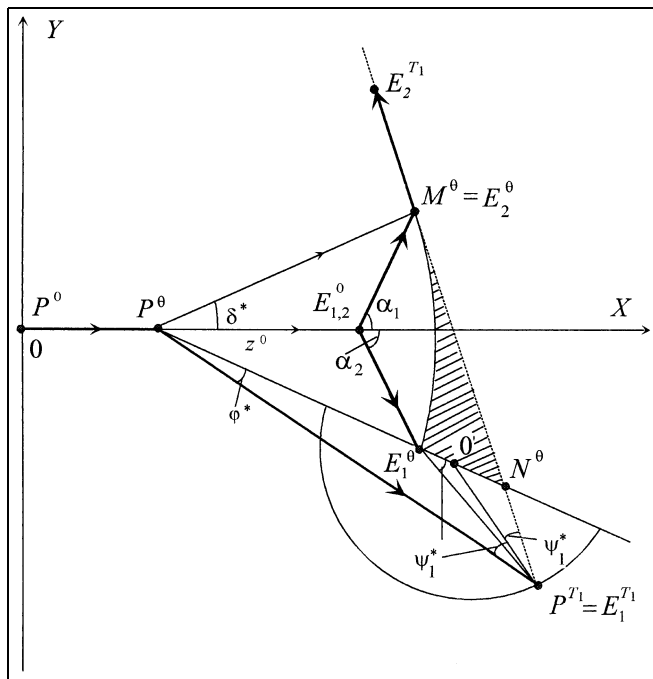


Рис. 6. Структура оптимальных траекторий при позиционном выборе очередности встреч с целями

Расчеты показывают, что при  $0,4 \leq \beta \leq 0,8$  позиционный выбор очередности позволяет на 10–20% сократить длину траектории преследования.

## 2. ДИФФЕРЕНЦИАЛЬНЫЕ ИГРЫ В СМЕШАННЫХ СТРАТЕГИЯХ

Дифференциальные игры в смешанных стратегиях являются математическими моделями задач противодействия подвижных объектов, в которых уклоняющаяся сторона осуществляет подавляющее воздействие на информационные системы противника. Однако одновременно подавляются и информационные системы уклоняющегося. В отсутствие текущей информации о фазовых координатах противника игроки используют программные управления до момента восстановления контакта. В этом состоит основная содержательная особенность задач управления в условиях подавляющего воздействия средств РЭП.

Для синтеза оптимальных управлений подвижными объектами в условиях подавляющего воздействия был применен аппарат теории программных игр в смешанных стратегиях [18, 19].

Приведем в качестве примера постановку и решение плоскостной дифференциальной игры для случая, когда истинная цель  $E_1$  использует для подавления информационной системы преследователя  $P$  неподвижную ложную цель  $E_2$ .

Вводится неподвижная система координат  $XOY$ , начало которой совпадает с начальным положением преследователя  $P^0$ , а ось абсцисс проходит через общее начальное положение целей  $E_{1,2}^0$ . Преследователь снаб-

жен зоной обнаружения  $\Omega[w_p(t)]$ , имеющей вид круга (кругового сектора), центр которого совпадает с текущим положением преследователя  $w_p(t)$  в системе координат  $XOY$ . Начальным считается момент постановки ложной цели. Предполагается, что после постановки ложной цели преследователь  $P$  и истинная цель  $E_1$  движутся по прямолинейным траекториям с постоянными скоростями. Направления их движений задаются углами  $\varphi$  и  $\psi$ , которые составляют прямолинейные траектории преследователя  $P$  и истинной цели  $E_1$  с осью абсцисс, соответственно.

Оптимальные стратегии игроков ищутся в классе смешанных. В качестве элементов чистых стратегий выбираются их программные управления (траектории). Смешанные стратегии определяются как вероятностные распределения на множествах чистых стратегий.

Критерий образуется следующим образом. Определяется множество

$$T(\tilde{w}_1, \tilde{w}_p) = \{t: w_1(t) \in \Omega[w_p(t)]\}$$

моментов времени, в которые текущее положение истинной цели  $E_1$  оказывается внутри области обнаружения преследователя. Здесь  $\Omega[w_p(t)]$  — открытая область, совпадающая с зоной обнаружения преследователя  $P$  в момент  $t$ ;  $w_1(t)$  и  $w_p(t)$  — двумерные векторы положений истинной цели  $E_1$  и преследователя  $P$ , соответственно, в момент  $t$ ;  $\tilde{w}_1, \tilde{w}_p$  — соответствующие траектории объектов.

Критерий — вероятность уклонения цели  $E_1$  от обнаружения преследователем  $P$  — есть математическое ожидание платежной функции

$$G = Pr\{T(\tilde{w}_2, \tilde{w}_p) = \emptyset\} = M_\chi\{T(\tilde{w}_2, \tilde{w}_p)\} \rightarrow \min_{P_\varphi} \max_{E_\psi} \quad (11)$$

где  $P_\varphi$  и  $E_\psi$  — вероятностные распределения на множествах чистых стратегий (направлений движения) игроков  $P$  и  $E_1$ , соответственно;  $M$  — символ математического ожидания; платежная функция имеет вид:

$$\chi\{T(\tilde{w}_2, \tilde{w}_p)\} = \begin{cases} 0, & T(\tilde{w}_1, \tilde{w}_p) \neq \emptyset \\ 1, & T(\tilde{w}_1, \tilde{w}_p) = \emptyset. \end{cases}$$

Этот критерий минимизируется по управлениям преследователя  $\varphi$  и максимизируется по управлениям истинной цели  $\psi$ , что и отражается в формуле (11).

Решение задачи ищется в смешанных стратегиях. При нахождении оптимальных стратегий используется прием построения “редчайшей” сети [18, 19].

В решаемой задаче этот прием сводится к построению траекторий участников группы, состоящей из минимального числа  $N$  действующих совместно преследователей, обладающих тем свойством, что при любом законе уклонения истинная цель  $E_1$  будет обнаружена хотя бы одним преследователем.

В результате решения задачи определяются:

— множество оптимальных поисковых траекторий преследователя  $P$ ; это множество состоит из траекторий участников указанной группы поисковых средств;



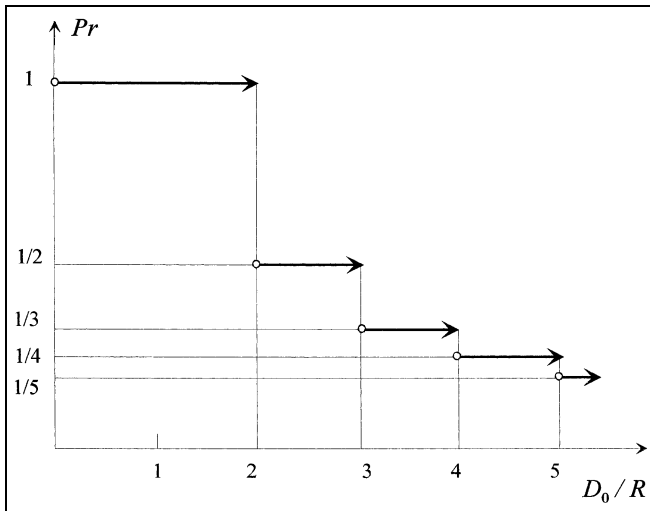


Рис. 7. График цены игры в смешанных стратегиях

— множество оптимальных траекторий уклонения истинной цели  $E_1$ ; траектории из этого множества обладают тем свойством, что на каждой траектории уклоняющаяся цель может быть обнаружена не более, чем одним преследователем из группы;

— цена игры — вероятность  $G^*$  уклонения цели  $E_1$  от обнаружения преследователем.

В работе [18] показано, что при прямолинейных траекториях преследования—уклонения и круговой (секторной) области обнаружения преследователя множество оптимальных чистых стратегий игроков состоит ровно из  $N$  элементов, а оптимальными смешанными стратегиями являются равномерные дискретные распределения на множествах чистых стратегий.

На рис. 7 приведен график зависимости  $Pr = 1 - G^*$  — вероятности  $R$ -встречи игроков (обнаружения истинной цели преследователем) от отношения  $R/D_0$ , где  $D_0$  — начальное расстояние между объектами,  $R$  — радиус круговой зоны обнаружения преследователя. График построен для случая, когда скорость цели в два раза меньше скорости преследователя.

## ЗАКЛЮЧЕНИЕ

Выделен новый класс задач конфликтного управления подвижными объектами, названный авторами "управление в условиях искусственно организованной неполноты информации". Содержательно сформулированные задачи описаны математическими моделями управления подвижными объектами в условиях имитирующего и подавляющего воздействий средств радиоэлектронного подавления информационных систем противника.

Для оптимизации законов управления в условиях имитирующего воздействия был разработан новый раздел теории дифференциальных игр преследования—уклонения — дифференциальные игры с групповой целью. Для оптимизации законов управления в условиях подавляющего воздействия привлечен математический аппарат дифференциальных игр в смешанных стратегиях.

Рассмотрены, в основном, результаты, полученные в Институте проблем управления им. В. А. Трапезникова РАН.

Исследования в описанных в статье направлениях ведутся и в других исследовательских центрах: в России — университетах Санкт-Петербурга и Владивостока, на Украине — в Институте кибернетики НАНУ, а также — в Израиле, США, Англии, Германии.

## ЛИТЕРАТУРА

1. Радиоэлектронная борьба в ВМФ / Под ред. Ю. В. Ефимова. — М.: Оружие и технологии, 2003. — 244 с.
2. Олышанский В. К., Рубинович Е. Я. Простейшие дифференциальные игры преследования системы из двух объектов // Автоматика и телемеханика. — 1974. — № 1. — С. 24–34.
3. Маслов Е. П., Рубинович Е. Я. Дифференциальные игры преследования-уклонения с групповой целью // Итоги науки и техники. Техническая кибернетика. — 1991. — Т. 32. С. 32–58.
4. Абрамянц Т. Г., Маслов Е. П. Дифференциальная игра преследования групповой цели // Известия РАН. Теория и системы управления. — 2004. — № 5. — С. 16–22.
5. Шевченко И. И. Стратегии сближения с коалициями в целом. — Владивосток: Изд-во Дальневост. ун-та, 2004. — 132 с.
6. Шевченко И. И. О сближении с коалицией // Автоматика и телемеханика. — 1986. — № 1. — С. 47–55.
7. Шевченко И. И. О поочередном преследовании // Автоматика и телемеханика. — 1981. — № 11. — С. 54–59.
8. Шевченко И. И. Поочередное преследование трех убегающих // Автоматика и телемеханика. — 1983. — № 7. — С. 70–75.
9. Абрамянц Т. Г., Маслов Е. П., Рубинович Е. Я. Простейшая дифференциальная игра поочередного преследования // Автоматика и телемеханика. — 1980. — № 8. — С. 5–15.
10. Иванов М. Н., Маслов Е. П. О сравнении двух методов преследования в задаче о поочередной встрече // Автоматика и телемеханика. — 1983. — № 8. — С. 38–43.
11. Шевченко И. И. Геометрия альтернативного преследования. — Владивосток: Изд-во Дальневост. ун-та, 2004. — 240 с.
12. Иванов М. Н., Маслов Е. П. Об одной задаче уклонения // Автоматика и телемеханика. — 1984. — № 8. — С. 56–62.
13. Об одной задаче уклонения от обнаружения / Т. Г. Абрамянц, М. Н. Иванов, Е. П. Маслов, В. П. Яхно // Автоматика и телемеханика. — 2004. — № 10. — С. 3–12.
14. Айзекс Р. Дифференциальные игры. — М.: Мир, 1967. — 415 с.
15. Петросян Л. А., Ширяев В. Д. Групповое преследование одним преследователем нескольких преследователей // Вестник ЛГУ. — 1980. — № 13. — С. 50–57.
16. Чикрий А. А., Калашиникова С. Ф. Преследование управляемым объектом группы убегающих // Кибернетика. — 1987. — № 4. — С. 1–8.
17. Shevchenko I. I. Successive Pursuit with a Bounded Detection Domain // Journal of Opt. Theory and Appl. — 1997. — Vol. 95, N 1.
18. Ермолов А. Н., Кряковский Б. С., Маслов Е. П. Об одной дифференциальной игре в смешанных стратегиях // Автоматика и телемеханика. — 1986. — № 10. — С. 32–45.
19. Железнов В. С., Кряковский Б. С., Маслов Е. П. Простейшая дифференциальная игра с выбором момента времени в смешанных стратегиях // Автоматика и телемеханика. — 1987. — № 8. — С. 46–55.

☎ (095) 334-91-81

E-mail: @ipu.ru



# АЛЕКСАНДР МИХАЙЛОВИЧ ПЕТРОВСКИЙ

## (к 80-летию со дня рождения)

В марте 2005 г. исполнилось бы 80 лет со дня рождения доктора технических наук, профессора Александра Михайловича Петровского, заведующего лабораторией управления по неполным данным Института проблем управления РАН.

Александр Михайлович Петровский родился 17 марта 1925 г. в семье московской научной интеллигенции. Его дед, А. Г. Петровский, был до революции главным санитарным врачом г. Москвы. Отец Александра Михайловича, М. А. Петровский, и дядя, Ф. А. Петровский, были известными филологами, переводчиками французской и античной литературы. Воспитанием Александра Михайловича занималась мать. Она и определила его путь “технаря”, считая, что стране необходимы технические кадры.

После окончания Московского энергетического института, с 1949 г. и до конца своей жизни А. М. Петровский работал в Институте автоматики и телемеханики АН СССР (ныне — Институт проблем управления РАН им. В. А. Трапезникова), пройдя путь от младшего научного сотрудника до заведующего основанной им же лабораторией управления по неполным данным.

А. М. Петровский вел исследования в разных направлениях. Одно было неизменным — неполнота информации об управляемой системе.

Начиная с конца 1950-х гг., А. М. Петровский исследует и разрабатывает способы повышения эффективности управляемых снарядов, наводимых на маневрирующие цели. Исследования охватывали как разработку алгоритмов наведения, так и построение моделей маневрирования подвижных объектов. Для этого широко применялись средства аналоговой вычислительной техники. Впервые в мировой практике для моделирования поведения маневрирующих целей им было предложено применить математический аппарат условных марковских процессов. Результаты исследований составили докторскую диссертацию, защищенную в 1966 г.

Дальнейшим развитием стали выполненные под руководством А. М. Петровского исследования, связанные с формализацией и решением задач противодействия подвижных объектов в условиях искусственно организованной неполноты информации. Указанные задачи возникают в широко распространенных на практике ситуациях, когда одна из участвующих в конфликте сторон применяет для срыва действий противника средства радиоэлектронного противодействия. Искусственная неполнота информации организуется путем постановки ложных целей, имитирующих характеристики основного



объекта, и (или) подавлением каналов наблюдений. Было предложено моделировать противодействие подвижных объектов дифференциально-игровыми и оптимизационными задачами преследования-уклонения групповой цели и дифференциально-игровыми и оптимизационными задачами поиска активно противодействующих подвижных объектов (поиск в условиях конфликта).

В середине 1960-х гг. А. М. Петровский начал заниматься новым для того времени объектом теории управления — медико-биологическими системами. Начав с поисков объекта управления в отдельных системах организма — кровообращение, терморегуляция, водно-солевой обмен, Александр Михайлович вскоре ини-

цировал работы по применению методов классической теории управления в биологических системах. Его лаборатория начала многолетние исследования в области моделирования гомеостаза организма для управления искусственным сердцем. Позже это направление было дополнено исследованиями в других инженерно-физиологических системах — работами в области искусственной терморегуляции у человека в сложных условиях окружения (в космосе или под водой) и работами по моделированию искусственных внутренних органов, таких как искусственная почка. Кроме того, А. М. Петровский занимался проблемой лечения — под его руководством проводились исследования способов подбора различных комбинаций лекарственных препаратов с учетом их взаимодействия и изучались проблемы влияния временных дозировок препаратов на онкологические процессы.

Исследования процессов управления в медико-биологических системах, проводимые под руководством А. М. Петровского, развивались в двух направлениях — с одной стороны, это была проблема моделирования и управления медико-биологическими объектами на уровне отдельного организма, с другой, — управление здоровьем (системой здравоохранения) на уровне популяции (всего населения). В 1989 г. им (в соавторстве с А. И. Михальским и А. И. Яшиным) была опубликована основополагающая монография “Теория оценивания неоднородных популяций”. Специальное внимание было уделено изучению системы здравоохранения как объекта управления. Под руководством А. М. Петровского методами системного анализа были разработаны математические модели системы охраны здоровья населения и разработаны методы комплексного учета показателей ее работы.



Выбор направления исследований оказался правильным. Проблема управления сложными биологическими системами и процессами в них является в настоящее время одним из наиболее перспективных направлений в теории управления. В современных мировых индексах цитирования первые двадцать мест безраздельно отданы научным журналам в этой области.

Проблематика управления здравоохранением в течение ряда лет была и остается востребованной как на государственном уровне, так и международным научным сообществом. Работы по этой и смежным тематикам велись широким фронтом сотрудниками лаборатории в рамках проекта «Народонаселение» Международного Института прикладного системного анализа, (г. Лаксенбург, Австрия) и продолжают в настоящее время в рамках международного сотрудничества.

Многие годы А. М. Петровский был заместителем заведующего кафедрой «Техническая кибернетика» (ныне «Проблемы управления») Московского физико-технического института. Многие из руководимых им студентов и аспирантов стали докторами и кандидатами наук.

Александр Михайлович запомнился всем его знавшим своей добротой, открытостью и человечностью. В созданной им лаборатории царил дружелюбная и теплая атмосфера. Это проявлялось как в совместной работе, так и на отдыхе. Александр Михайлович любил шутку и веселый розыгрыш. Даже выезды на овощную базу или колхозное поле — столь частые в советские вре-

мена — он умел превратить в веселое приключение. Своим оптимизмом и жизнерадостностью он заражал окружающих. У него не было недоброжелателей — он всегда умел конструктивно разрешить возникающие проблемы.

А. М. Петровский был человеком редкостной порядочности. В годы его работы в Институте категорически не поощрялись переходы сотрудников из одной лаборатории в другую. Дирекция предпочитала расстаться с обоими участниками конфликта. И только А. М. Петровскому удалось сохранить в Институте ряд способных исследователей, по разным причинам пришедшихся «не ко двору» в других лабораториях. Среди них такие известные ныне ученые, как В. Н. Вапник, А. Я. Черво-ненкис, Р. Ш. Липцер и др.

Лаборатория и сегодня сохраняет легкость и непри-нужденность отношений, свойственные ей во времена Александра Михайловича.

А. М. Петровский скончался по пути на работу 1 февраля 1993 г. Но созданная им лаборатория живет и успешно продолжает его дело. Научные идеи А. М. Петровского по-прежнему актуальны и развиваются его учениками и учениками учеников. И это, наверное, лучший памятник Ученому.

*Е. П. Маслов*

☎ (095) 334-91-81

E-mail: maslov@ipu.ru



## НАУЧНЫЕ ЧТЕНИЯ ПАМЯТИ А. М. ПЕТРОВСКОГО

В Институте проблем управления им. В. А. Трапезникова РАН 17 марта 2005 г. состоялись научные чтения, посвященные памяти профессора Александра Михайловича Петровского (1925—1993). Их тематика была связана с работами Александра Михайловича и охватывала исследования по управлению движущимися объектами, обучающимся машинам и алгоритмам обучения, а также работы в области биологических, здравоохраненческих и медико-социальных применений теории управления. В научных чтениях приняли участие российские и зарубежные ученики и коллеги А. М. Петровского.

На открытии чтений выступили директор Института проблем управления академик Грузинской академии наук *И. В. Прангшвили*, директор Института проблем передачи информации РАН академик РАН *Н. А. Кузнецов*, заведующий кафедрой информатизации при Президиуме Российской академии медицинских наук профессор *Д. Д. Венедиктов* и заведующий лабораторией Института проблем управления РАН д-р техн. наук *Е. П. Маслов*.

В своем вступительном слове *И. В. Прангшвили* охарактеризовал А. М. Петровского как ученого, внесшего большой вклад в формирование облика Института, сложившегося в 1960-е гг., когда Институт возглавлял научные направления, связанные с теорией управления. В эти годы в Институте работала целая плеяда выдающихся ученых, в которую вместе с группой молодых тогда специалистов входил и А. М. Петровский. Эти ученые

создали основы теории управления и начали работы по применению развитых методов в различных практических областях — от управляемых зенитных ракет и управления движущимися объектами до биологических применений теории управления и создания искусственного сердца.

Интересными воспоминаниями о работах А. М. Петровского в области управления зенитными ракетами поделился академик *Н. А. Кузнецов*. Труды Александра Михайловича были связаны с разработкой нелинейных алгоритмов управления ракетами и со временем привели к созданию систем управления наблюдениями. Результаты исследований проверялись на реальных объектах, в том числе и во Вьетнаме. О результативности отечественных разработок того времени можно судить по следующему факту. Советские ракетные системы применялись, в частности, для защиты моста через р. Меконг, по которому проходила известная «тропа Хо Ши Мина». Разрушение этого моста не позволило бы северовьетнамцам проникать на Юг, но американцы так и не смогли разбомбить его.

Своими воспоминаниями об А. М. Петровском поделился профессор *Д. Д. Венедиктов* (в 1970-х гг. — заместитель министра здравоохранения СССР и председатель Научного совета по системному анализу). Он рассказал о работах А. М. Петровского в области математического моделирования системы управления здра-

вохранением. По его словам, система здравоохранения в СССР была в то время одной из самых передовых в мире по организации управления и по численности врачей, хотя в области техники и уступала западному здравоохранению. При активном участии А. М. Петровского в Международном институте системных исследований (IIASA) в Вене было создано научное направление по математическому моделированию системы здравоохранения. Это направление работ в течение ряда лет успешно развивалось. Д. Д. Венедиктов подробно рассказал о работе над этим проектом и высоко оценил участие в нем самого А. М. Петровского, а также А. И. Яшина, А. А. Клементьева и других сотрудников ИПУ.

Е. П. Маслов, сменивший А. М. Петровского в качестве заведующего лабораторией № 38 Института проблем управления, рассказал о работах этой лаборатории. Обладая широтой взглядов, А. М. Петровский охотно принимал в состав своей лаборатории те группы сотрудников, которые по тем или иным причинам оставались без научного руководства. Сам Е. П. Маслов пришел в эту лабораторию вместе с группой сотрудников, продолжившей после смерти А. А. Фельдбаума исследования по статистическим методам управления. Позже в лабораторию влились сотрудники А. Я. Лернера, эмигрировавшего из СССР. Они продолжили исследования по распознаванию образов и применили эти методы в биологии и медицине. Еще одной группой, принятой в лабораторию, были сотрудники Н. В. Позина, занимавшиеся исследованием биологических систем. Александр Михайлович курировал эти направления, давая всем сотрудникам возможность продолжать исследования и помогая им. Недаром докторами наук в лаборатории стало значительное число сотрудников (всего за прошедшие годы докторские диссертации защитили десять человек).

После открытия Чтений состоялись доклады, в которых рассматривались проблемы управления подвижными объектами, алгоритмы управления и обучения и проблемы моделирования популяционных и биологических систем.

Первым бы заслушан доклад *Т. Г. Абрамянц, Е. П. Маслова и Е. Я. Рубиновича* (ИПУ) “Управление подвижными объектами в условиях искусственно организованной неполноты информации”. В нем были рассмотрены задачи конфликтного управления подвижными объектами в условиях радиоэлектронного противодействия (РЭП). Подробнее см. статью в настоящем номере журнала, с. 75–81.

*В. А. Шлаин*, бывший аспирант Института проблем управления, представляющий ныне компанию “Microspec Technologies” (Израиль), сделал доклад об алгоритмах автоматического обнаружения и классификации дефектов при производстве полупроводниковых схем. Субъективность в определении типа дефектов, низкая производительность и высокая стоимость анализа микросхем, производимого человеком-оператором, диктует необходимость автоматизации этого процесса. В фирме “Microspec Technologies” разработана система DCS-4, расширяющая возможности рабочих станций, предназначенных для визуальной инспекции результатов технологического процесса (например, “Axiospect 200” производства фирмы “Carl Zeiss”). Обнаружение дефектов в этой системе проводится в три этапа. На первом этапе изображение совмещается с референтным (не содержащим дефекта) изображением и конструируется маска

дефекта. При этом применяются методы улучшения изображения (фильтрация, сглаживание, увеличение контрастности границ деталей изображений), восстановления изображения (зуммирование, ротация), регистрации (совмещения) изображений, производятся сегментация и бинаризация изображений и их морфологический анализ. На втором этапе вычисляются числовые признаки дефекта и его окружения на микросхеме. К этим признакам относятся морфологические признаки (размеры и форма дефекта); признаки локализации дефекта по отношению к границам и сегментам, содержащим различные материалы; признаки, определяющие материал и цветовые характеристики дефекта; признаки, относящиеся к текстуре дефекта. На третьем этапе определяется тип дефекта — классификация. Основные функции системы DCS-4 заключаются в классификации дефектов, выполняемой с помощью методов распознавания образов, и интерактивной поддержке пользователя в определении новых классов дефектов, выполняемой с помощью методов кластерного анализа и специальной лингвистической поддержки.

Для решения задачи классификации дефектов в системе DCS-4 используется концепция мультиклассификации, заключающаяся в применении нескольких методов классификации в зависимости от соотношения количества априорной и статистической информации. В случае, если обучающая выборка отсутствует, но имеется априорная информация о предполагаемых дефектах, применяется алгоритм классификации, основанный на теории нечетких множеств. В случае, если обучающая выборка мала и имеется априорная информация о предполагаемых дефектах, применяется алгоритм автоматической генерации правил классификации, базирующийся на основе логики нечетких множеств. Впоследствии пользователь может вручную корректировать классификационные правила вручную в соответствии с имеющейся у него априорной информацией. В случае, если обучающая выборка достаточно велика для статистического анализа, применяется полностью автоматическая генерация правил классификации, основанная на методе ближайшего соседа, автоматически отбираются наиболее информативные признаки.

Система DCS-4 снабжена специальной интерактивной подсистемой поддержки пользователя при определении новых классов дефектов, которая осуществляет следующие функции:

- с помощью семантической поддержки пользователь может выбрать подмножество классификационных признаков, перспективных для разделения множества неизвестных ранее дефектов на классы;
- на основе этих признаков дефекты автоматически разделяются на кластеры;
- для каждого полученного кластера (множества дефектов, не имеющих меток принадлежности к классам) автоматически порождается вербальное описание, характеризующее средние значения выбранных признаков в кластере.

Пользователь может проанализировать полученные вербальные описания, вручную перераспределить объекты между кластерами, создать новые классы на основе полученных кластеров или произвести новую кластеризацию на основы иного набора признаков.

Затем был заслушан доклад *В. Н. Буркова* (ИПУ РАН) “Теория активных систем — современное состоя-





ние и перспективы”. Необходимость учета так называемого “человеческого фактора” в задачах управления организационными системами была осознана специалистами по управлению в 1960 гг. Под человеческим фактором понимается активное влияние человека (коллектива, организации) как объекта управления на систему управления. Это активное влияние проявляется, в основном, в двух аспектах. Один из них связан с так называемой проблемой манипулирования, когда объект управления сознательно представляет в систему управления искаженные данные о своем состоянии и возможностях. Другой связан со способностью объекта управления сознательно ограничивать область допустимых состояний. В конце 1960 гг. возникла теория активных систем, объектом управления в которой выступал активный элемент, характеризующийся отмеченными свойствами.

Основная проблема теории активных систем заключается в согласовании целей управляющей системы и активных элементов путем синтеза управляющих воздействий (механизмов управления), обеспечивающих как достоверность получаемой от активных элементов информации, так и их функционирование “на границе возможностей”. К настоящему времени в теории активных систем получен ряд фундаментальных результатов по синтезу оптимальных механизмов управления.

Доклад *И. Н. Воронцова* (ИПУ) был посвящен проблемам управления в междисциплинарных исследованиях. В течение ряда лет реализуется научная программа исследований по выбору концептуального базиса системы, который позволяет описывать и исследовать сложные явления различных типов. За это время создана система математических методов машинного моделирования, ориентированная на применение в широком диапазоне исследовательских и проектных работ. Концептуальные основы созданного языка для описания и анализа систем различных типов успешно прошли многоступенчатую проверку.

Трудности решения междисциплинарных проблем (таких, как проблема устойчивого развития или управление процессами в сложных системах большой размерности) связаны с тем, что лабильность биологических и социально-экономических систем часто очень велика. Быстрые изменения в этих системах приводят к невозможности получения сведений о значениях переменных, необходимых для реализации законов управления. В лабильных варьируемых системах управление возможно только при выявлении достаточно стабильных закономерностей как инвариантов лабильности.

Предложенный подход к преодолению возникающих трудностей основан на уменьшении размерности подлежащих анализу моделей: декомпозиции их описания на одном семантическом уровне. Другой подход основан на представлении модели системы в виде иерархии подсистем на разных семантических уровнях. Такое решение возможно только при условии существования общего концептуального базиса и языка описания свойств системы. После того, как были выполнены исследования возможности интерпретации предикатов и предметных имен выражений естественного языка в терминах языка системы моделирования, задача создания фреймовой базы данных для понимания текстов конкретных научных дисциплин стала вполне разрешимой.

Актуальная задача, решение которой открывает принципиально новые возможности в биологии и меди-

цине, состоит в создании математической модели эукариотической клетки и конструировании компьютерной модели ее функционирования, интегрирующей данные по геному, протеомиксу и другим клеточным системам. Основная особенность этой задачи заключается в автоматизации процессов синтеза моделей на основании сведений, заложенных в трех основных типах баз данных о метаболизме: сведений о метаболических путях, ферментах и соответствующих реакциях; сведений о метаболитах и, наконец, физико-химических закономерностях кинетики реакций.

Успехи разработки такой модели будут зависеть от успехов в создании методов измерения концентраций метаболитов *in situ*, от создания общих методов синтеза и исследования моделей биологических процессов и от развития работ по онтологии биологических процессов, генома и клетки.

*М. С. Сергеев* (Московский физико-технический институт), *А. П. Серебровский* (Институт передачи информации РАН) и *Г. Ш. Цицашвили* (Институт прикладной математики ДВО РАН) сделали доклад, в котором рассмотрели нестандартный метод оценки параметра сигнала на фоне шумов. Была рассмотрена задача оценки параметра произвольного сигнала, наблюдаемого в смеси с аддитивным гауссовским шумом в дискретные моменты времени. Традиционно при решении таких задач применяют либо разложение в комплексный ряд Фурье, либо метод наименьших квадратов. Однако для реализации первого метода требуются наблюдения за достаточно длительным промежутком времени, больший или сравнимый с периодом принимаемого сигнала. Если условия задачи этого не позволяют, авторы предлагают применять метод наименьших квадратов, который для достижения необходимой точности требует большего объема вычислений. Предлагаемый в работе метод основан на декомпозиции результатов наблюдений, что позволяет на порядок снизить объем вычислений. Его применение оказалось эффективным в приложениях к задачам акустики при оценке параметров самых разнообразных нелинейных волн.

*Н. А. Коргин* (ИПУ РАН) выступил с сообщением “Механизмы обмена как основа распределения научной и учебной нагрузок преподавателей”. В докладе различные виды взаимодействия между субъектами социально-экономических систем рассматриваются как обмен между ними, приносящий выигрыш каждой из обменивающихся сторон. Взаимодействие между руководством кафедры вуза и профессорско-преподавательским составом (ППС) рассматривается как пример описанной ситуации. При этом процесс распределения учебной и научной нагрузок можно трактовать как обмен в следующем виде: кафедра раздает имеющийся у нее в наличии ресурс (время на различные виды деятельности) и взамен получает результат деятельности преподавателей. Этот подход позволяет построить математическую модель кафедры как обменной схемы и применить имеющиеся на данный момент результаты из теории управления организационными системами и теории активных систем, полученные при решении различных задач обмена, в частности, задач стимулирования. Источником для математической модели кафедры служит инструкция по нормированию основных видов деятельности профессорско-преподавательского и научно-го составов Академии Федеральной службы охраны РФ.

В этой инструкции определены объёмы годовой нагрузки по различным видам работ для ППС (бюджеты времени), примерные нормы времени на различные виды деятельности, а также закреплено право руководства кафедры на перераспределение бюджетов времени между ППС в определенных пределах.

Кафедра рассматривается как двухуровневая организационная система, на верхнем уровне которой находится руководство кафедры (центр), а на нижнем — преподаватели кафедры (активные элементы). В рамках доклада моделируется исключительно процесс распределения научной нагрузки. Решается задача повышения результативности научной деятельности кафедры в различных условиях информированности руководства кафедры о параметрах системы, необходимых для принятия эффективных решений.

В последующих докладах были рассмотрены проблемы моделирования популяционных и биологических систем.

Доклад *В. К. Иванова* (Национальный радиационно-эпидемиологический регистр, г. Обнинск) был посвящён формированию групп потенциального риска. Важность этой задачи обусловлена двумя основными обстоятельствами. Во-первых, в результате выполнения крупномасштабных эпидемиологических исследований медицинских последствий радиационного воздействия (Хиросима, Нагасаки, Чернобыль, Южный Урал, регистры атомных работников ведущих стран мира и др.) стало возможным дать объективную оценку фактора риска радиационной природы на индивидуальном уровне. Во-вторых, наличие эпидемиологических данных об индивидуальных радиационных рисках позволяет оптимизировать систему радиационной защиты и минимизировать отдаленные эффекты радиационного воздействия с использованием технологии оказания адресной медицинской помощи через страховые компании.

В докладе показано решение задачи формирования групп потенциального риска на примере больших контингентов жителей Российской Федерации, подвергшихся радиационному воздействию вследствие чернобыльской катастрофы. Кроме того, в анализируемый контингент включен персонал атомной промышленности страны, находящийся на индивидуальном дозиметрическом контроле.

В соответствии с Постановлением Правительства РФ № 948 от 22 сентября 1993 г. создан и успешно функционирует Национальный радиационно-эпидемиологический регистр. На 1 января 2005 г. в нем имеются персональные медико-дозиметрические данные на 615 тыс. человек, в том числе на 186 тыс. участников ликвидации последствий чернобыльской катастрофы (“ликвидаторов”) и 368 тыс. жителей наиболее загрязненных радионуклидами областей России.

Один из основных выводов постчернобыльских эпидемиологических исследований, подтвержденных Всемирной организацией здравоохранения (ВОЗ), говорит о существенном повышении частоты заболеваемости раком щитовидной железы у населения в наибольшей степени загрязненных радионуклидами областей России (Брянская, Калужская, Тульская и Орловская). Общая численность этого населения составляет 5,6 млн. чел., заболеваемость раком щитовидной железы среди жителей указанных территорий выросла в 3—4 раза. Эти фактические данные о частоте заболеваемости раком щито-

видной железы могут быть объяснены двумя основными факторами: нерадиационной природы — скрининг и радиационной природы — инкорпорированное облучение щитовидной железы радионуклидами йода-131. Выполненные Национальным радиационно-эпидемиологическим регистром крупномасштабные исследования к настоящему времени показали, что к группе потенциального радиационного риска можно отнести только детское (0—14 лет) на момент чернобыльской катастрофы население юго-западных районов Брянской области (это 26 908 человек, проживающих в 362-х населенных пунктах). Персональные списки этого контингента переданы органам практического здравоохранения для оказания эффективной медицинской помощи.

Национальным радиационно-эпидемиологическим регистром также завершена работа по анализу частоты заболеваемости лейкозами среди ликвидаторов. Установлено, что к группе потенциального риска следует отнести примерно 28 тыс. ликвидаторов, которые работали в 1986 г. и получили дозы внешнего облучения более 150 мЗв.

Национальный радиационно-эпидемиологический регистр в настоящее время начал радиационно-эпидемиологические исследования по формированию группы потенциального риска среди персонала атомной отрасли, находящегося на индивидуальном дозиметрическом контроле. Предварительные данные (по Ленинградской АЭС) свидетельствуют о том, что численность этой группы составляет примерно 10 % всего персонала, размер средней накопленной дозы — 215 мЗв. Представленные в докладе данные включены в итоговые отчеты ВОЗ. Они отражены в ряде монографий на русском и английском языках и опубликованы в ведущих отечественных и международных научных журналах, в том числе в регулярно издающемся Бюллетене Национального регистра “Радиация и риск”.

Затем был заслушан доклад *Г. П. Иткина* (НИИ трансплантологии и искусственных органов Минздрава РФ) “Программа искусственного сердца в России”. Сердечно-сосудистые заболевания уносят больше человеческих жизней, чем все остальные заболевания, вместе взятые, и от этой патологии ежегодно умирает более 100 млн. жителей Земли. Несмотря на значительные средства, вкладываемые в фармакологическую индустрию, наиболее радикальным средством помощи больным с тяжелыми формами сердечной недостаточности является пересадка сердца. Однако дефицит донорских органов делает практически невозможным оказание помощи всем больным, нуждающимся в подобной операции. В листе ожидания в странах мира ежегодно находятся 25 тыс. больных и только менее 5 % из них получают донорское сердце. Большинство реципиентов умирают, не дождавшись донорского сердца. В этих случаях единственная возможность спасти жизнь — подключить к организму искусственное сердце или вспомогательный насос на период ожидания донорского сердца.

К настоящему времени имплантируемые системы искусственного сердца и вспомогательного кровообращения нашли применение во многих клиниках мира. Однако высокая стоимость этих систем (от 150 000 до 400 000 долл. США) практически исключают возможность их применения в нашей стране, а потребность в таких системах оценивается по минимуму в пределах 100 систем в год. И в дальнейшем, по мере приобрете-



ния клинического опыта, эта потребность будет увеличиваться в десятки раз.

Поэтому для решения этой проблемы была разработана новая государственная программа *Искусственное сердце*, к которой были привлечены более 10 малых предприятий, финансируемых Государственным Фондом развития малого бизнеса в научно-технической сфере. Основная цель программы заключалась в разработке и создании комплекса новых методов и средств в области кардиохирургии и трансплантологии для лечения больных с тяжелыми формами сердечной недостаточности.

При составлении двухлетней программы *Искусственное сердце* были выделены основные направления, позволяющие при относительно небольших финансовых затратах получить максимальную отдачу. При этом часть выделенных средств была направлена на развитие новых технологий, также предназначенных для диагностики и нормализации сердечной деятельности.

В результате выполнения двухлетней программы *Искусственное сердце* были разработаны опытные образцы следующих аппаратов:

- имплантируемый электромеханический пульсирующий насос, неппульсирующие центробежный и аксиальный насосы для длительного подключения к организму на период ожидания донорского сердца;
- одноразовый центробежный насос для временного подключения к организму (до одного месяца) при невозможности отключения аппарата “сердце—легкие” после кардиохирургических операций;
- автоматический дефибриллятор в комплексе с аппаратом для наружного массажа сердца;
- ультразвуковой внутритриповодный сканер для ранней диагностики сердечной недостаточности;
- комплекс биодegradуемых матриц для кардиомиоцитов и нервных стволовых клеток.

В настоящее время разработанные образцы новой техники проходят стендовые и медико-биологические испытания и в ближайшие годы будут внедрены в клиническую практику кардиохирургических и реанимационных центров страны.

Выступление *А. И. Михальского* (ИПУ РАН) было посвящено рассмотрению прямых и обратных задач в анализе популяций. Докладчик очертил круг процессов и индикаторов, характеризующих состояние популяции людей: заболеваемость, смертность, распространение в популяции факторов риска. Рассматривая эти процессы в рамках модели стохастических переходов индивидуумов между различными состояниями здоровья и факторов риска, можно математически выразить связь между вероятностными характеристиками на индивидуальном уровне и характеристиками, наблюдаемыми на уровне популяции. В результате удаётся выразить ожидаемые изменения в наблюдаемой в популяции заболеваемости при изменении заболеваемости индивидуума. Эти соотношения важны при оценке результатов воздействия на популяцию и дают пример решения прямой задачи — описания процесса от *причины* к *следствию*. Другие примеры решения прямых задач касались оценки результатов профилактики, влияния изменения численности групп риска на заболеваемость и смертность в популяции, что важно при анализе последствий техногенных катастроф.

Специфика обратных задач заключается в необходимости построения оценок процесса (который является *причиной*) по наблюдению за *результатом*. Примерами

являются оценка заболеваемости по данным о смертности, оценка заболеваемости по числу выявленных случаев заболевания (распространённость заболевания) и т. д. Актуальность такого подхода вытекает из того, что многие важные состояния здоровья и переходы между ними не поддаются непосредственному наблюдению, а регистрируются лишь конечные терминальные состояния. В докладе было показано, как обратные задачи анализа популяции формально свести к задаче решения систем интегральных уравнений первого рода, связывающих функции распределения и интенсивности переходов наблюдаемых и ненаблюдаемых процессов. Обсуждались принципиальные трудности решения таких систем, связанные с их некорректностью, и пути преодоления возникающих трудностей. Был приведён пример решения реальной задачи оценки онкологической заболеваемости лиц, принимавших участие в ликвидации последствий аварии на Чернобыльской АЭС в 1986 г. Применение методов решения обратных задач позволило эффективно скорректировать влияние пропусков в наблюдениях состояния здоровья этих людей, возникающих вследствие нерегулярности медицинских обследований. Стабилизация соответствующих оценок позволила получить обоснованную динамику развития онкологических заболеваний среди указанного контингента.

В докладе был поднят вопрос о возможности повышения достоверности решения обратных задач путём рассмотрения процессов, по отношению к которым оцениваемые процессы являются *результатом*. Так, процессы в популяции являются *результатом* по отношению к процессам, протекающим на уровне индивидуума. Однако процессы в индивидууме являются *результатом* процессов, протекающих на уровне клетки и обусловленных как окружающей средой, так и генетически. По этой причине изучение генетических свойств популяции может позволить по популяционным данным получить уточнённые оценки процессов на индивидуальном уровне. Этим обстоятельством объясняется большой интерес к популяционным генетическим исследованиям (например, различные проекты Европейского сообщества), что, в свою очередь, выдвигает требования к развитию новых методов решения обратных задач, в частности, с применением новых статистических подходов.

*Н. А. Бабушкина* (ИПУ РАН) посвятила своё выступление моделированию эффективности действия противоопухолевых препаратов в сверхмалых дозах. Его основные положения отражены в статье, опубликованной в настоящем номере журнала (см. с. 47—54).

*Л. А. Дартау* и *А. Р. Стефанюк* (ИПУ РАН) выступили с докладом “Компьютерная технология и система ЭДИФАР: этапы развития”, в котором осветили историю, принципы и задачи создания экспертной диалоговой информационной системы для анализа рисков. Компьютерная система ЭДИФАР была создана для автоматизированного сбора данных о распространенности жалоб и факторов риска, касающихся основных хронических неинфекционных заболеваний, определяющих в последние десятилетия преждевременную смертность и инвалидизацию населения развитых стран. Эта работа была инициирована *А. М. Петровским* в 1985 г. в рамках академической темы “Исследование неоднородных популяций и управление по неполным данным”. Испытания системы прошли на базе одной из поликлиник Москвы. За прошедшие 20 лет результаты работы выли-



лись в создание теоретических и прикладных основ управления здоровьем на популяционном уровне и обоснование возможности реализации разрабатываемого подхода в условиях Российской Федерации. Сформулированы три основных вида деятельности государства в сфере охраны здоровья населения, которые, помимо эпизодического лечения граждан, предусматривают периодический скрининг и перманентное обучение здоровью. Работа поддержана отечественными и международными грантами, доложена на научно-практических конференциях в России и за рубежом, её результаты изложены в многочисленных публикациях<sup>1</sup>.

Доклад *А. С. Колоколова* (ИПУ РАН) под названием “Обработка сигнала в частотной области при распознавании речи” был посвящен преобразованиям логарифмического кратковременного спектра речевого сигнала, учитывающим особенности анализа звука в слуховой системе. Использование таких преобразований в процессе обработки речи снижает вариации спектра сигнала, обусловленные характеристиками среды и речевым источником. Эффективность рассмотренных преобразований подтверждается примерами их применения при обработке реальных речевых сигналов.

Первое преобразование выделяет локальные неоднородности логарифмического спектра по частоте, которыми являются максимумы и резкие срезы спектра, и позволяет получить частотное описание речевого сигнала, устойчивое к частотным искажениям сигнала и широкополосным аддитивным шумам. Этим обеспечивается более надёжное сравнение с эталонными описаниями речевых фрагментов при распознавании речи. В основу преобразования положены разновидности полосовой фильтрации спектра по частоте, используемая для нахождения второй сглаженной разности спектра, и последующая нелинейная обработка, имитирующая эффект латерального торможения в слуховом анализаторе.

Второе преобразование выделяет локальные неоднородности спектра по времени и может быть применено для сегментации речевого сигнала. Оно представляет собой вариант полосовой фильтрации спектра, реализующей нахождение сглаженной первой разности логарифмической огибающей спектра по времени. В результате преобразования отмечаются моменты начала и конца артикуляторных событий. Такое преобразование спектра имеет место в слуховом анализаторе, о чём свидетельствует присутствие так называемых тонических нейронов, избирательно реагирующих на начало и конец акустического стимула. На основе этого преобразования предложены устойчивые к частотным искажениям процедуры сегментации речевого сигнала для выделения в нём последовательных квазистационарных сегментов, необходимых для фонетической разметки.

Возможные области применения рассмотренных преобразований — распознавание речи, идентификация и верификация диктора.

*А. Я. Червоненкис* (ИПУ РАН) сделал сообщение “Применение методов распознавания образов в задачах

молекулярной биологии” (см. статью в настоящем номере журнала, с. 41—46).

В заключительном докладе *В. Н. Новосельцева* (ИПУ РАН) “Моделирование гомеостаза: история и современное состояние” говорилось о развитии гомеостатических представлений в биологии. Первоначальные идеи *К. Бернара* и *И. М. Сеченова* о постоянстве внутренней среды в середине XX в. привели к развитию концепции гомеостаза. Концепция была создана *У. Кенноном*, а *Н. Винер* представил модель гомеостаза в виде системы управления с отклонением от уставки (например, в системе регуляции температуры вводилась уставка по температуре). Однако быстро выяснилось, что такая модель не отражает всей сложности гомеостатического поведения биологических систем. Например, самка питона, обычно пойкилотермная (не обладающая постоянством температуры тела), в период высиживания яиц приобретает гомеостатические свойства. Мало того, она поддерживает гомеостаз только в определенной области внешних температур, за пределами которой температура тела самки не держится на постоянном уровне, а следует за температурой внешней среды.

Эти сложности удалось преодолеть в новой модели гомеостаза, созданной в 1978 г. В качестве главной цели системы управления в биосистемах было выдвинуто равенство темпов потребления веществ и их поступления. Только после того, как эта цель достигалась, в системе возникал гомеостаз — относительное постоянство уровня (концентрации) веществ. Уставки на концентрации в этой модели отсутствовали. Такая модель не только воспроизводила механизм гомеостаза в ограниченной области изменения окружающих условий, но и моделировала гомеостаз при изменении темпов потребления веществ. Последнее обстоятельство оказалось важным при моделировании процессов старения биосистем.

В настоящее время одной из самых распространенных и достоверных теорий старения считается теория окислительных повреждений (*Д. Харман*, 1957). Ее суть состоит в том, что нормально потребляемый организмом кислород в качестве побочного эффекта порождает необычайно активные свободные радикалы, с возрастом поражающие все структуры организма. Математическая модель этой теории, разработанная докладчиком в 2000—2001 гг., демонстрирует экспоненциально замедляющееся с возрастом снижение гомеостатической способности. Такое сокращение приводит к возрастному падению кислородного ресурса, а достижение им нулевого уровня приводит организм к смерти — отсутствие ресурсов не допускает восстановления гомеостаза даже при сколь угодно малых нарушениях. Созданная на основе этих рассуждений общая схема жизненного цикла организма дала возможность исследовать многие аспекты старения и стала основой ряда публикаций в зарубежных и отечественных журналах.

Чтения прошли на высоком научном уровне, а сделанные доклады вызвали большой интерес научной общественности и привлекли внимание учёных и специалистов из Москвы и других городов.

*А. И. Михальский,  
В. Н. Новосельцев*

☎ (095) 334-88-20, 334-88-91

E-mail: [mikhalsk@ipu.ru](mailto:mikhalsk@ipu.ru)

[Novoselc@ipu.ru](mailto:Novoselc@ipu.ru)



<sup>1</sup> См., например, статью *Дартау Л. А.* “Теоретические аспекты управления здоровьем и возможности его реализации в условиях Российской Федерации”. — Проблемы управления. — 2003. — № 2. — С. 43—52.