ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

6(154) 2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу http://www.**informika.ru/text/magaz/it/** или **http://novtex.ru/IT.**

Журнал включен в базу данных Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.4'414

А. Е. Ермаков, канд. техн. наук,
рук. отдела компьютерной лингвистики,
e-mail: ermakov@rco.ru,
В. В. Плешко, ген. директор, e-mail: vp@rco.ru,
ООО "ЭР СИ О" (www.rco.ru)

Семантическая интерпретация в системах компьютерного анализа текста

Описывается подход к построению семантического компонента в системах компьютерного анализа текста на естественном языке. Подход основан на применении специальных шаблонов к сети синтактико-семантических отношений между словами текста, которая строится синтаксическим анализатором. Шаблоны определяют способ интерпретации фрагментов сети в заданные фреймы с идентификацией участников ситуаций и их ролей.

Ключевые слова: *компьютерный анализ текста, семантическая интерпретация, семантическая сеть, синтаксический анализ, фреймы.*

Введение

К числу современных компьютерных систем, использующих машинный анализ текста на естественном языке, традиционно относятся информационно-поисковые и вопросно-ответные системы, автоматические переводчики, а также широкий класс так называемых систем извлечения знания из текста, среди которых выделяют системы KM (*knowledge management*) и BI (*business intelligence*), системы сбора фактографической информации и ведения конкурентной разведки. Во всех подобных системах ключевым этапом автоматической обработки текста является семантическая интерпретация выражений естественного языка на основании определенной семантической модели мира (предметной области), результатом которой является формирование формальных структур, инвариантных к несодержательным лексико-грамматическим особенностям написания текста автором и соответствующих требованиям решаемой прагматической задачи.

Существуют различные подходы к построению семантического компонента систем анализа текста.

Первый подход, который следовало бы назвать "сильным", изначально возникший в рамках работ по машинному переводу, предполагает использование специальных семантических метаязыков для описания значения предложения [4]. Основоположниками этого подхода в нашей стране являются Ю. Д. Апресян [1] и И. А. Мельчук [2], а среди их последователей сегодня особенно выделяется коллектив проф. В. А. Тузова, практическая работа которого подробно описана в работе [3]. В рамках этого подхода каждое значение слова должно быть описано семантической формулой, а множество всех таких описаний представляет собой семантический словарь языка. Например, одно из значений слова *потушить* описывается формулой *Perf Caus*(*im, Fin Lab*(*vin, ОГОНЬ*)), где *огонь* — это базисное, далее не разложимое понятие, а *Perf, Caus, Fin, Lab* — это базисные семантические функции. Так, базисная функция *Lab*(*x*, *y*) означает, что аргумент, обозначаемый словом *x*, подвергается действию аргумента, обозначаемого словом *y*. Значение предложения описывается математическим выражением. Например, предложению *Пожарники потушили загоревшийся сарай* соответствует выражение *Perf Caus* (*ПОЖАРНИКИ, Fin Lab*(*Perf Incep Labo 1*(*САРАЙ, ОГОНЬ*), *ОГОНЬ*)), которое в обратном переводе на русский язык имеет следующее значение: *Пожарники сделали так, что перестал подвергаться действию огня начавший подвергаться действию огня сарай.* Задача создания семантического метаязыка заключается в выборе системы таких базисных функций и понятий, которые позволяют описать значения всех других понятий и предложений, дальнейшее толкование которых либо невозможно, либо нецелесообразно. Так, метаязык, описанный в работе [3], содержит 72 базисные функции и около 600 базисных понятий.

Будучи изначально созданным для целей автоматического перевода текста с одного языка на другой, "сильный" подход к интерпретации текста на основе семантических метаязыков вынужденно претендует как на полноту охвата множества интерпретируемых выражений естественного языка, так и на точность (подробность) семантического описания этих выражений, что вызывает множество проблем при попытках практической реализации адекватного метаязыка [4], ни одна из которых по сей день не увенчалась успехом.

Прочие подходы, более "слабые", но вместе с тем более прагматичные, изначально подразумевают интерпретацию только тех фрагментов текста, которые описывают искомые отношения между сущностями предметной области или ситуации, в которые вовлечены эти сущности, не претендуя при этом на точность семантического описания. Например, приведенное выше предло-

жение может быть проинтерпретировано просто как связь "*пожарник → тушить пожар → сарай*" или как ситуация "*пожар: объект = сарай*", в зависимости от назначения прикладной системы. К этой группе можно отнести самые разные методы семантической интерпретации, используемые в системах извлечения знаний из текста, вплоть до самых слабых, вообще не учитывающих синтаксис языка.

Статья посвящена методу семантической интерпретации, разработанному в компании "ЭР СИ О" (http://www.rco.ru). В соответствии с приведенной классификацией метод относится к группе "слабых", однако по факту полноты использования лингвистической информации всех уровней он является наиболее "сильным" в этой группе, во всяком случае, подобных методов автору не известно ни в России, ни за рубежом. Описываемый семантический компонент определяет, каким образом будут интерпретированы те или иные языковые конструкции, описывающие те или иные ситуации, в заданные фреймы, с идентификацией участников ситуации и их ролей. С этой целью для каждого типа ситуаций создаются особые синтактико-семантические шаблоны, позволяющие распознать и проинтерпретировать допустимые способы описания ситуации в тексте. Такие шаблоны применяются не к тексту, а к сети синтактико-семантических отношений между словами, которая строится синтаксическим анализатором, что обеспечивает высокую инвариантность шаблонов к особенностям поверхностно-синтаксической организации предложений и, как следствие, ускоряет разработку лингвосемантического описания предметной области.

Сеть синтактико-семантических отношений

В результате синтаксического анализа предложения и последующих трансформаций дерева синтаксических зависимостей между словами [5] формируется сеть синтактико-семантических отношений — семантическая сеть. Семантическая сеть содержит все сущности, упоминавшиеся в тексте предложения: наименования предметов и лиц, действий и признаков, связанные различными типами синтактико-семантических связей. Направление связи обычно соответствует направлению синтаксического подчинения слов. Пример семантической сети представлен на рис. 1.

Узлы и связи в сети имеют набор следующих основных атрибутов:

- часть речи слова, соответствующего узлу (*SpeechPart*);
- семантический разряд референта узла (*SemanticType*), например, *персона, организация, гео-*

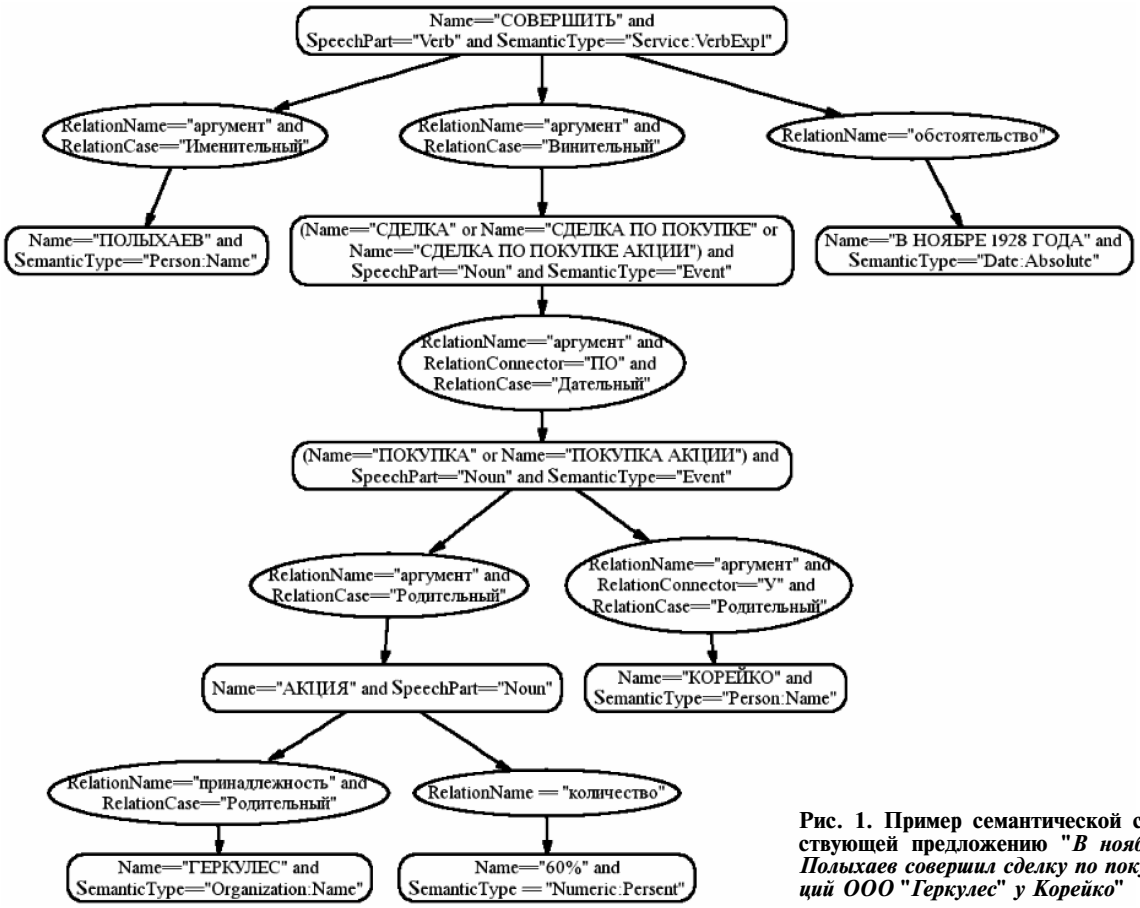


Рис. 1. Пример семантической сети, соответствующей предложению "*В ноябре 1928 года Полыхаев совершил сделку по покупке 60 % акций ООО "Геркулес" у Корейко*"

графическое место, действие/состояние, предмет и др.;

- строка текста, соответствующая узлу, в нормальной форме (*Name*). Для именных групп она может иметь несколько значений, которые представляют все цельные словосочетания, образованные от главного существительного в узле, например, *сделка по покупке акции, сделка по покупке, сделка*;
- тип синтактико-семантической связи между узлами (*RelationType*), например, *аргумент* (*совершить → сделку*), *принадлежность* (*акция → Геркулеса*), *обстоятельство* (*совершить → в 1928*);
- семантический падеж (*RelationCase*) и коннектор (*RelationConnector*) − предлог или союз, с помощью которого устанавливается связь. Комбинации условий *RelationConnector* + *RelationCase* представляют альтернативу традиционным семантическим ролям (*субъект, объект, инструмент, локатив* и т. п.), инвентарь которых в лингвистике так и не определен окончательно. При этом формально различные грамматические падежи слов в тексте, выбор которых определяется поверхностно-синтаксической организацией фразы, отображаются в один и тот же семантический падеж. Например, семантический именительный падеж субъекта действия и винительный объекта действия соответствуют в активном залоге одноименным грамматическим падежам (*программист написал программу*), в пассивном залоге выражаются грамматическим творительным и именительным падежами соответственно (*программистом написана программа*), а в причастном обороте вообще могут выражаться любыми грамматическими падежами (программисту, написавшему...; о программе, написанной...).

Семантическая сеть инвариантна к синтаксической структуре и порядку слов с точностью до структуры пропозиции, выбранной автором для описания ситуации. Например, конструкциям *"Корейко купил акции Геркулеса у Берлаги*" и "*акциями Геркулеса, купленными Корейко у Берлаги*" будут соответствовать одинаковые сети. В то же время пропозициям вида "*Корейко становится покупателем акций Геркулеса*" и "*покупка акций Геркулеса — дело рук Корейко*" будут соответствовать иные сети. Описанная семантическая сеть является промежуточным уровнем представления между собственно семантической схемой ситуации и ее конкретным языковым описанием, т. е. представлением глубинно-синтаксического уровня, абстрагированным от особенностей поверхностного синтаксиса.

Фреймы и семантические шаблоны

Логическая схема ситуации в терминологии искусственного интеллекта называется фреймом. Фрейм имеет имя, которое идентифицирует тип описываемых им ситуаций (например, *купля-продажа акций*), а также содержит слоты, которые имеют свои имена, идентифицирующие роли участников ситуации. Для конкретной ситуации часть слотов может быть заполнена именами ее конкретных участников, упомянутых в тексте (*покупатель = Корейко, эмитент акций = Геркулес, количество акций = 60 %, дата = 1928, продавец = ?, сумма сделки = ?*).

Для семантической интерпретации каждого способа описания ситуации в тексте используется соответствующий синтактико-семантический шаблон. На рис. 2 приведен пример такого шаблона, соответствующего пропозиции вида *Покупатель совершает действие по приобретению акций предприятия-эмитента у продавца*.

Как видно, синтактико-семантический шаблон задается в виде сети, изоморфной искомой в тексте сети, в узлах и связях ее с помощью логических выражений указываются условия, которым должны удовлетворять узлы и связи искомой сети. Обычно в некоторых узлах шаблона содержатся конкретные слова, которые должны присутствовать в тексте. Другие узлы — валентности шаблона — соответствуют искомым участникам и дополнительно содержат обозначения их ролей — это имена слотов фрейма, заполняемых словами из текста при нахождении фрагмента семантической сети, соответствующего шаблону. Так, на рис. 2 узлы с именами *Buyer, Issuer, Seller и Share* представляют возможных участников ситуации "*покупка акции*" в ролях "*Покупатель*", "*Эмитент акций*", "*Продавец*", "*Размер доли*" соответственно. Светлые связи к фигурантам *Seller, Issuer и Share* помечены как факультативные, так как соответствующие участники могут не упоминаться в тексте.

В рамках описываемой модели семантическая интерпретация описания ситуации в тексте есть поиск в его семантической сети такой подсети, которая изоморфна шаблону, с заполнением слотов соответствующего фрейма именами участников ситуации из текста в соответствии с ролями, указанными в узлах шаблона.

На практике возможны такие случаи, когда синтаксический анализатор не может установить связь между словами, опираясь на заложенные в него общие правила грамматики, например, в таких текстах особого стиля: *Соучредитель ООО "Геркулес"* (20 %) *— ЗАО "Рога и копыта"*. Чтобы решить такие проблемы, в семантическую сеть добавляются связи особого типа (*RelationType* = = "*next*"), которые связывают в цепочку идущие друг за другом в предложении слова и знаки препинания, причем "перепрыгивая" через синтаксически подчиненные слова в именных группах и связывая только их вершины, что позволяет писать шаблоны, инвариантные к числу слов в словосо-

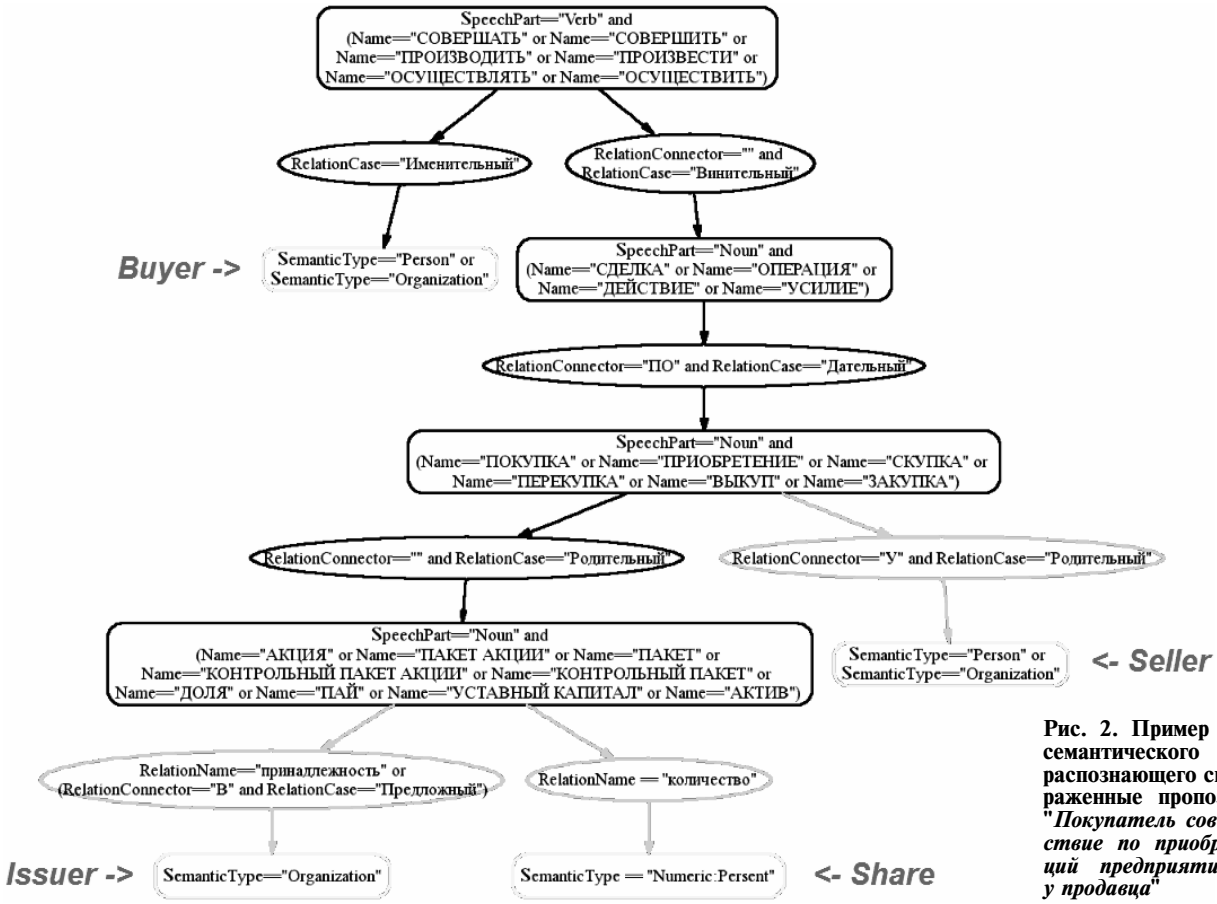


Рис. 2. Пример синтактико-семантического шаблона, распознающего ситуации, выраженные пропозицией вида "*Покупатель совершает действие по приобретению акций предприятия-эмитента у продавца*"

четаниях. В итоге совокупность узлов сети всегда представляет собой полносвязный граф, что позволяет описывать на основании единого формализма как универсальные общеязыковые, так и специфические для предметной области и стиля текста типы текстовых конструкций.

Расширенная семантическая интерпретация

Для уменьшения числа шаблонов, описывающих целевые фреймы, существует возможность создавать служебные шаблоны, назначение которых состоит не в заполнении фреймов, а в добавлении к семантической сети текста дополнительных узлов и связей с заданными атрибутами. В ходе семантической интерпретации все шаблоны применяются в определенном порядке, и каждый следующий шаблон обрабатывает сеть, которая является совместным результатом работы синтаксического анализатора текста и всех предыдущих шаблонов, содержащих порождаемые узлы и связи. Так, если для всех типов извлекаемых фреймов все конструкции, построенные по типу "*Субъект принимает решение действовать* (*о действии*) ...", предполагают такую же интерпретацию, как и конструкции типа "*Субъект действует*", то вместо добавления соответствующего шаблона для каждого типа фрейма целесообразно написать один служебный шаблон (рис. 3), который будет добавлять в сеть связь от названия действия к субъекту, вследствие чего подобные описания ситуаций будут распознаваться по простым шаблонам типа "*Субъект действует*".

Кроме того, для уменьшения числа создаваемых шаблонов компонент семантической интерпретации поддерживает средства их параметризации, которые позволяют подключать внешние словари к требуемым ролям шаблонов, указывая слова и семантические разряды допустимых или, наоборот, недопустимых слов, что позволяет выделять фреймы с одинаковой структурой слотов, но разными типами, на основании одного и того же множества общих шаблонов. Так, например, имея множество общих шаблонов, описывающих ситуацию "*приобретение-покупка чего-либо*", можно посредством подключения словарей к роли "*Товар*" определить фреймы типа *"приобретение предприятий*", "*приобретение акций*", "*приобретение недвижимости*", а также "*приобретение прочих вещей*", исключив из числа *прочих вещей* все *предприятия, акции, недвижимость,* а также *влияние*, *доверие* и им подобные непредметные сущности.

Ключевой особенностью текстов некоторых стилей (биографии, описания товаров) является высокая плотность таких связей между словами,

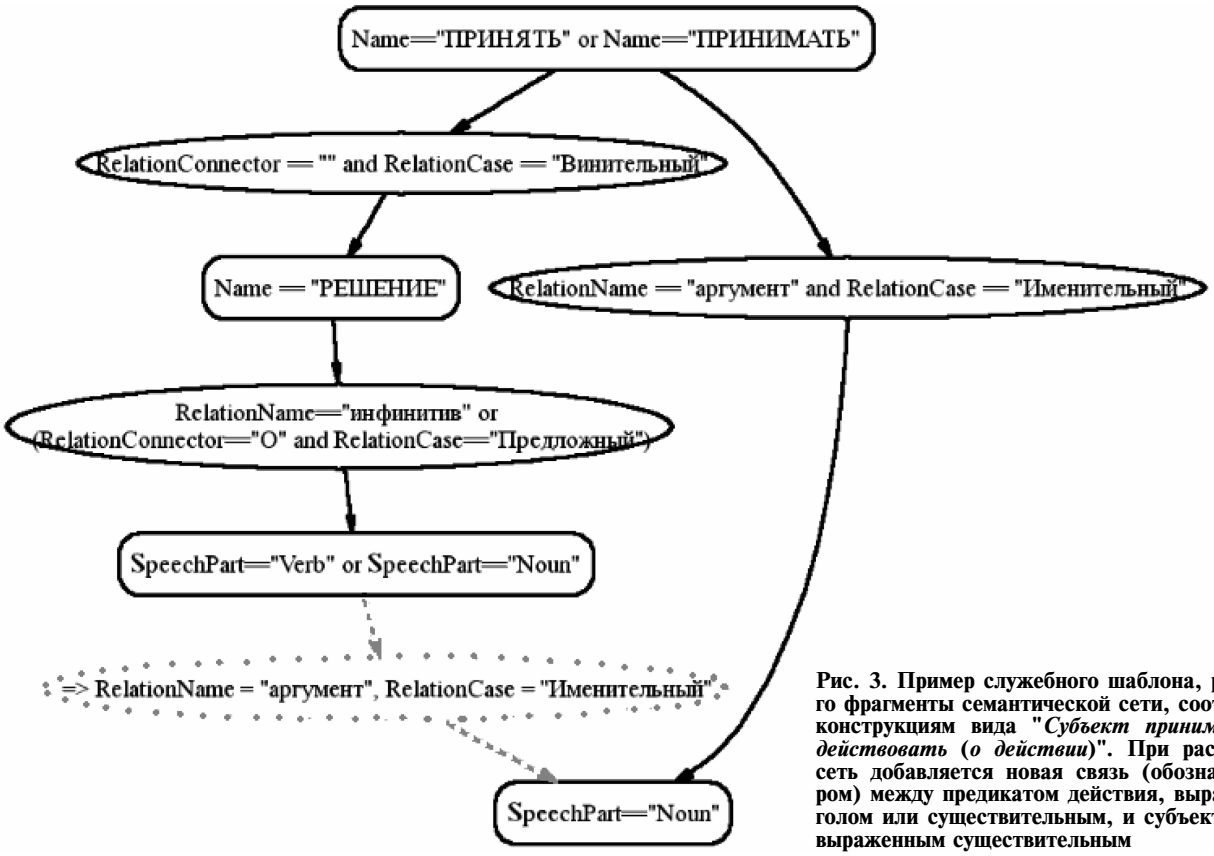


Рис. 3. Пример служебного шаблона, распознающего фрагменты семантической сети, соответствующие конструкциям вида "*Субъект принимает решение действовать* (*о действии*)". При распознавании в сеть добавляется новая связь (обозначена пунктиром) между предикатом действия, выраженным глаголом или существительным, и субъектом действия, выраженным существительным

которые не выражаются грамматическими средствами, — анафорических связей. Большинство предложений в подобных текстах либо бессубъектно (*Родился в 1958 году. Работает директором ООО "Геркулес"*), либо номинативно (*1958 года рождения. Директор ООО "Геркулес"*), либо вообще разорвано в списках (*Является владельцем акций следующих предприятий: — ООО "Геркулес", основано в 1928 году .... — ООО "Рога и копыта", основано в 1924 году ... — ООО "Ударник", основано в 1925...*). Для поиска тех факультативных участников ситуации, которые не были найдены в предложении в результате применения шаблона, разработан специальный механизм поиска анафорических связей по всему тексту, опирающийся на специальные лингвистические правила и соответствующую маркировку узлов шаблонов [6].

Наконец, настройки семантического интерпретатора позволяют фильтровать описания тех ситуаций, которые соответствуют реальным событиям или фактам, на основании наличия в тексте общеязыковых показателей отрицания и нереальности (*не, якобы, если* и т. п.). В итоге из числа найденных могут быть исключены те фреймы, которые соответствуют нереальным ситуациям (*Корейко купил бы акции "Геркулеса"*), и те участники, которые реального участия в ситуации не принимали (*Корейко купил акции не "Геркулеса"*).

Настройка семантических шаблонов

Семантические шаблоны задаются на формальном языке описания графов DOT (http://www.graphviz.org). Для удобства их разработки используется приложение с графическим интерфейсом, которое строит сеть на основе типовой фразы естественного языка, т. е. обучает семантический компонент на примерах. В итоге граф шаблона строится автоматически, после чего человеку остается проставить необходимые логические выражения в узлы (обычно требуется добавление синонимов), указать роли искомых участников ситуации, пометить обязательных и факультативных участников. Окончательно шаблон сохраняется в формализме DOT, готовый для загрузки компонентом семантической интерпретации, который обеспечивает поиск изоморфизмов и заполнение фреймов по семантической сети при анализе текста.

Для автоматизированного построения большого количества семантических шаблонов используется следующий метод их пакетного построения.

1. Языковая основа каждого шаблона описывается типовой конструкцией на естественном языке с соблюдением следующих соглашений:

- типовая конструкция состоит из известных слов естественного языка и представляет собой грамматически правильную фразу. Каждое слово конструкции может представлять собой макроопределение — отсылать к одноименно-

му классу слов, способных занимать это словоместо. При этом синтаксическая форма конструкции однозначно определяет типы связей, устанавливаемых между соответствующими узлами в графе шаблона;

- макроопределение может быть описано непосредственно в составе конструкции или вынесено в отдельную запись и имеет одну из четырех допустимых форм: *Макроопределение = слово1 = = слово2 = ... = словоN* или *Макроопределение = = { слово1, слово2, словоN }* или *Макроопределение = [логическое выражение, допустимое в графе шаблона]* или *Макроопределение:РольУчастника*. Первые три формы транслируются в логическое выражение и определяют ограничения на узел шаблона. Последняя форма транслируется в роль участника ситуации, с возможным маркером его факультативности "~", например: *Продавец:~Seller*. Во избежание возможных конфликтов, вызванных многозначностью слов, макросы могут иметь глобальную и локальную области видимости.

Пример двух типовых конструкций с макросами, описывающих два способа выражения ситуации покупки акций:

```
Покупатель:Buyer; Покупатель =
{SemanticType == "Person" or SemanticType ==
"Organization"};
Продавец:~Seller; Продавец = {SemanticType ==
"Person" or SemanticType == "Organization"};
Эмитент:~Issuer; Эмитент = {SemanticType ==
"Organization"};
Покупка = {приобретение, скупка, перекупка,
выкуп, закупка};
Покупать = {купить, приобретать, приобрести,
скупать, скупить...};
Акции = {пакет, пакет акций, контрольный пакет,
контрольный пакет акций, доля, пай, уставный
капитал, актив},
Покупатель Покупает Акции Эмитента у Продавца;
Покупательсовершает=производить=произвести=
осуществлять=осуществить сделку=операция=
действие=усилие по Покупке Акций Эмитента
у Продавца;
```

2. Пакет типовых конструкций обрабатывается программой-транслятором, которая подвергает синтаксическому анализу каждую конструкцию, строя соответствующую ей семантическую сеть — граф шаблона, и подставляя вместо слов — обозначений макросов — соответствующие им определения. В ходе трансляции создается журнал, в котором фиксируются следующие основные ошибки:

- ошибки разбора текста описания шаблонов, связанные с нарушением формата;
- ошибки или неоднозначности разбора типовой конструкции (полный синтаксический разбор не удался или допустимо несколько вариантов разбора);
- слово в типовой конструкции, написанное с большой буквы, не удалось отождествить с макросом. Возможно, соответствующий макрос не определен или нормальная форма слова не совпадает с макросом.

3. После анализа сообщений об ошибках эксперт-настройщик устраняет их одним из двух возможных способов:

- корректирует текст описания: исправляет синтаксические ошибки, добавляет определения макросов, переформулирует текст омонимичных конструкций, после чего вновь запускает программу-транслятор;
- корректирует непосредственно семантические шаблоны, используя графическое приложение.

Заключение

Описанный подход к семантической интерпретации внедрен в компонент лингвистического анализа текста RCO Fact Extractor и успешно апробирован на базе русского и английского языков, обеспечивая в среднем около 95 % точности и 60 % полноты при извлечении из текста описаний событий и фактов в соответствии с заданными семантическими шаблонами. В настоящий момент для русского языка уже разработано более 600 шаблонов, которые покрывают более 70 типов ситуаций, связанных с экономической и общественно-политической деятельностью персон и организаций. Структура фреймов (типы ситуаций, состав ролей участников) и соответствующие шаблоны разрабатывались "с нуля" на основе анализа текстовых примеров, предоставляемых заказчиками. Для английского языка, работа с которым недавно начата, в качестве полезного ресурса предполагается использовать базу данных проекта FrameNet (http://framenet.icsi.berkeley.edu/), которая содержит разработанные структуры более чем 800 фреймов, с примерами описаний соответствующих ситуаций более чем в 130 тысячах типовых предложений.

Список литературы

1. **Апресян Ю. Д.** Лексическая семантика. М.: Наука, 1974. 366 с.
2. **Мельчук И. А.** Опыт теории лингвистических моделей "Смысл ↔ Текст". М.: Наука, 1974.
3. **Тузов В. А.** Компьютерная лингвистика. Опыт построения компьютерных словарей. СПб.: Изд-во СПбГУ, 2002. 650 с.
4. **Кобозева И. М.** Лингвистическая семантика. М.: Эдиториал УРСС, 2000. 352 с.
5. **Ермаков А. Е.** Эксплицирование элементов смысла текста средствами синтаксического анализа-синтеза // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции Диалог'2003. М.: Наука, 2003. С. 136—140.
6. **Ермаков А. Е.** Автоматическое извлечение фактов из текстов досье: опыт установления анафорических связей // Компьютерная лингвистика и интеллектуальные технологии: Труды Международной конференции Диалог'2007. М.: Наука, 2007. С. 131—135.

УДК 004.82

И. А. Ходашинский, д-р техн. наук, проф.,
Томский государственный университет
систем управления и радиоэлектроники,
e-mail: hodashn@rambler.ru

Идентификация параметров нечетких моделей типа синглтон на основе алгоритма роящихся частиц

Описана идентификация параметров нечетких моделей типа синглтон на основе алгоритма роящихся частиц. Рассмотрены результаты имитационного эксперимента.

Ключевые слова: *идентификация нечетких моделей, метаэвристики, алгоритм роящихся частиц.*

Введение

Нечеткие системы находят широкое применение в таких проблемных областях, как распознавание образов и проектирование регуляторов, прогнозирование и моделирование, управление и принятие решений. Они встроены в такие потребительские товары, как фото- и видеокамеры, кондиционеры и холодильники, стиральные машины и др. К преимуществам нечетких систем относятся невысокая стоимость разработки, интуитивно понятная логика функционирования, гибкость. Однако применяются такие системы без должного обоснования выбора структуры и параметров модели. Указанные задачи должны быть решены на этапе идентификации модели. Основными задачами идентификации нечеткой модели являются выбор типа нечеткой модели, задание входных и выходных переменных, определение структуры функций принадлежности, выбор числа правил, идентификация параметров антецедентов и консеквентов правил, определение критериев оценки функционирования нечетких моделей.

Известны два подхода к идентификации нечетких моделей: на основе знаний эксперта и на основе таблиц наблюдений [1].

Проблема идентификации параметров нечетких моделей на основе таблиц наблюдений может быть решена как задача оптимизации, цель которой состоит в нахождении таких параметров модели, при которых ошибка между тестовыми и модельными значениями могла быть минимизирована. Тогда проблема сводится к проведению поиска в многомерном пространстве, координаты которого соответствуют параметрам нечеткой модели. В силу того что поверхность поиска в указанном пространстве имеет сложный рельеф, классические методы типа градиентного поиска здесь не всегда эффективны.

Для решения проблем оптимизации широко применяются метаэвристики [2]. Среди множества метаэвристик метод роящихся частиц выделяется простотой понимания и реализации, а также достаточно высокой вычислительной эффективностью. Алгоритм роящихся частиц — это средство поиска оптимума в пространстве непрерывно заданных параметров. Алгоритм предложили в 1995 г. инженер-электрик Russel Ebenhart и физиолог James Kennedy [3], основываясь на работах биолога Craig Reynolds, который, изучая социальное поведение птиц, получил формулу поведения птиц в стае. Этот метод широко применяется в научных и технических приложениях для решения задач оптимизации с непрерывно и дискретно меняющимися параметрами [4].

В нашей работе проведено экспериментальное исследование поведения алгоритма при идентификации нечетких моделей в зависимости от значений параметров алгоритма.

Постановка задачи

Нечеткая система, построенная на основе нечеткой модели, выступает в качестве универсального аппроксиматора. Построение аппроксимирующего описания объекта разделяется на следующие этапы:

- подбор структуры модели;
- оценивание параметров модели;
- выбор критерия качества аппроксимации и метода оптимизации выбранного критерия.

В работе рассматривается нечеткая система типа синглтон, в которой n входных переменных, m нечетких правил, каждое из которых имеет следующий вид:

$$\text{правило } j\text{: ЕСЛИ } x_1 = A_{1j}$$
$$\text{И } x_2 = A_{2j} \text{ И ... И } x_n = A_{nj} \text{ ТО } r_j,$$

где r_j — действительное число, $r_j \in \Re$. Нечеткая система осуществляет отображение F: $\Re^n \to \Re$, заменяя оператор нечеткой конъюнкции произведением, а оператор агрегации нечетких правил — сложением. Тогда выходное значение $F(x)$ вычисляется следующим образом:

$$F(\boldsymbol{x}) = \frac{\sum\limits_{j=1}^{m} r_j \prod\limits_{i=1}^{n} \mu_{Aij}(x_i)}{\sum\limits_{j=1}^{m} \prod\limits_{i=1}^{n} \mu_{Aij}(x_i)},$$

где $\boldsymbol{x} = [x_1, ..., x_n]^{\mathrm{T}} \in \Re^n$; $\mu_{Aij}(x_j)$ — функция принадлежности (ФП) нечеткого ПОНЯТИЯ A_{ij}.

Известно несколько типов функций принадлежности: треугольные, трапециевидные, гауссовы, параболические, каждая из которых задается определенным набором параметров. Без потери общности в работе рассматриваются треугольные функции принадлежности, для определения которых необходимо три параметра.

Основой для оценивания параметров функций принадлежности является таблица наблюдений либо тестовая функция $f(\boldsymbol{x})$.

Проблема идентификации нечетких моделей решается как задача оптимизации, цель которой состоит в том, чтобы найти параметры модели, при которых ошибка

$$\frac{\sum_{i}^{N}\left|f(\boldsymbol{x}_i)-F(\boldsymbol{x}_i)\right|}{N},\ \frac{\sqrt{\sum_{i}^{N}(f(\boldsymbol{x}_i)-F(\boldsymbol{x}_i))^2}}{N}$$

$$\text{или } \max_i(|f(\boldsymbol{x}_i)-F(\boldsymbol{x}_i)|) \qquad (1)$$

между тестовыми $f(\boldsymbol{x})$ и модельными значениями $F(\boldsymbol{x})$ могла быть минимизирована. В качестве такого средства в работе рассматривается алгоритм роящихся частиц.

Алгоритм роящихся частиц

Алгоритм роящихся частиц — это стохастический метод поиска, основанный на итеративном взаимодействии частиц, образующих рой. Каждая частица — это решение, заданное как координаты частицы в многомерном пространстве. Перемещение частицы в пространстве поиска определяют следующие три фактора: инерция, память, сотрудничество. Инерция подразумевает, что частица не может мгновенно изменить свое направление движения. Каждая частица имеет память и хранит свою лучшую позицию в пространстве поиска. Известна частице и лучшая позиция роя. Зная эти две позиции, частица динамически изменяет скорость согласно ее собственному опыту и опыту полета других частиц. Таким образом, движение каждой частицы задается ее лучшей позицией, ее текущей скоростью, ускорением, заданным предыдущей позицией, и ускорением, заданным лучшей частицей в рое. Рой прекращает движение при выполнении хотя бы одного из следующих условий:

- рой достиг состояния равновесия;
- найдено оптимальное решение (ошибка меньше заданной);
- выполнено определенное число итераций.

Пусть имеется n-мерное пространство поиска $S \subset \Re^n$, рой состоит из N частиц. Позиция i-й частицы определяется вектором

$$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in S.$$

Лучшая позиция, которую занимала i-я частица, определяется вектором

$$\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \in S.$$

Скорость частицы определяется также n-мерным вектором

$$\boldsymbol{v}_i = (v_{i1}, v_{i2}, \ldots, v_{in}).$$

Движение частиц определяют простые математические уравнения:

$$\boldsymbol{v}_i(k+1) = w\boldsymbol{v}_i(k) + c_1\text{rand}(\boldsymbol{p}_i(k) - \boldsymbol{x}_i(k)) + c_2\text{Rand}(\boldsymbol{p}_g(k) - \boldsymbol{x}_i(k));$$

$$\boldsymbol{x}_i(k+1) = \boldsymbol{x}_i(k) + \boldsymbol{v}_i(k+1),$$

где $i = 1, 2, \ldots, N$; $\boldsymbol{v}_i(k)$ — вектор скорости частицы i на итерации k; $\boldsymbol{x}_i(k)$ — координаты частицы i на итерации k; c_1, c_2 — положительные коэффициенты ускорения; $\boldsymbol{p}_i(k)$ — лучшая позиция частицы i на первых k итерациях; $\boldsymbol{p}_g(k)$ — лучшая позиция частицы в рое (задается индексом g) на первых k итерациях; w — эмпирический коэффициент инерции; rand, Rand — случайные числа из интервала [0, 1].

Коэффициент c_1 является когнитивным (познавательным) параметром, отражающим доверие частицы к ее собственному прошлому опыту, этот коэффициент ответственен за обнаружение новых областей в пространстве поиска. Коэффициент c_2 является социальным параметром, показывающим, насколько частица доверяет рою, этот коэффициент ответственен за исследование окрестностей ранее найденной перспективной области.

Коэффициент инерции ответственен за изменение скорости и управляет обнаружением новых областей и поиском в окрестностях перспективной области.

На рис. 1 показана общая концепция обновления решения i-й частицей в двухмерном пространстве поиска.

Процесс модификации решения повторяется многократно, пока не будет выполнено условие останова.

Пространство поиска S в задаче идентификации представлено всеми возможными параметрами нечеткой системы. На рис. 2 представлен век-

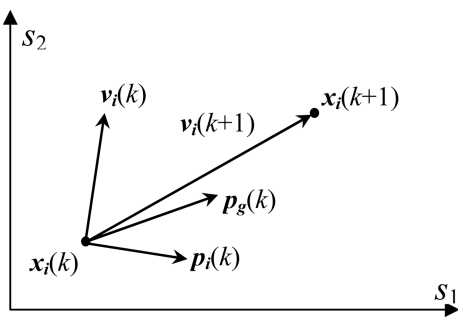


Рис. 1. Перемещение частицы в пространстве поиска

Ошибка	a_{11}	b_{11}	d_{11}	a_{12}	b_{12}	d_{12}	…	a_{24}	b_{24}	d_{24}	a_{25}	b_{25}	d_{25}

Рис. 2. Пример кодирования решения

тор, определяющий позицию *i*-й частицы, вместе с ошибкой, характеризующей данное решение, для треугольной функции принадлежности, заданной тремя параметрами (a, b, d), в двухмерном пространстве поиска, когда входные переменные представлены пятью терм-множествами:

Обобщенный алгоритм роящихся частиц:

Шаг 1. Задание параметров алгоритма: $\boldsymbol{v}_i(0) = 0$, c_1, c_2, w, N или минимальное значение ошибки.

Шаг 2. Генерация роя случайным образом, $k = 1$.

Шаг 3. Оценка каждой частицы роя путем вычисления ошибки по формулам (1). Если достигнуто условие окончания работы (число итераций, значение ошибки), то переход на шаг 6.

Шаг 4. Для каждой частицы определение ее вектора скорости $\boldsymbol{v}_i(k + 1)$ и новых координат $\boldsymbol{x}_i(k + 1)$.

Шаг 5. Расчет новых лучших позиций каждой частицы $\boldsymbol{p}_i(k + 1)$. Определение частицы с лучшей позицией в рое $\boldsymbol{p}_g(k + 1)$.

$k = k + 1$; переход к шагу 3.

Шаг 6. Окончание работы.

Эксперимент и обсуждение результатов

Для аппроксимации была выбрана следующая тестовая функция: $f(x_1, x_2) = x_1\sin(x_2)$. Начальное решение было одинаковым для всех экспериментов, критерием качества аппроксимации выступала максимальная ошибка, значение которой для начального решения равно 0,110724482.

Распределение максимальной ошибки для пяти опытов в зависимости от числа итераций и размера популяции приведено на рис. 3. Отметим, что здесь приведены не все проведенные нами опыты, а только наиболее типичные. Так, например, в одном из неприведенных опытов ошибка аппроксимации на 500 итерациях составила 4,092857719066E-10. Закодированное решение, соответствующее формату рис. 2, представлено на рис. 4.

В ходе эксперимента изучались индивидуальные траектории оптимизации при неизменном размере популяции, равном 20 особям, в зависимости от числа итераций. Различия в траекториях объясняются случайностью, присутствующей в определении вектора скорости.

Рассмотрим поведение алгоритма в зависимости от коэффициента инерции. Большие значения w направляют алгоритм на поиск новых перспективных областей, малые — на исследование окрестностей ранее найденных областей.

Коэффициент инерции меньше 1. При $0 < w < 1$ частицы замедляют движение, а скорость сходимости алгоритма определяется значениями коэффициентов c_1 и c_2.

При малых значениях c_1 и c_2 наблюдается преждевременная сходимость алгоритма, являющаяся следствием раннего обнуления скоростей частиц. На рис. 5 показано изменение максимальной ошибки для пяти испытаний при равных параметрах алгоритма. Здесь, так же как и в других приведенных ниже опытах, различия в траекториях объясняются случайностью, присутствующей в определении вектора скорости.

При $w = 0{,}3$, $c_1 = 1{,}4$, $c_2 = 1{,}4$ и популяции размером в 10 особей алгоритм сходится при 200—300 итерациях. При этом найденные решения далеки от оптимальных (ошибка решения равна 0,007—0,07).

При $c_1 > 0$ и $c_2 = 0$ каждая частица ведет независимый локальный поиск, никак не взаимодействуя с остальными частицами роя. Эффективность такого поиска очень низка, алгоритм сходится после нескольких десятков итерации, незначительно улучшая начальное решение. Указанное сочетание параметров непригодно для решения задачи идентификации параметров нечетких моделей.

При $c_1 = 0$ и $c_2 > 0$ все частицы исследуют окрестности найденного на предыдущих итерациях луч-

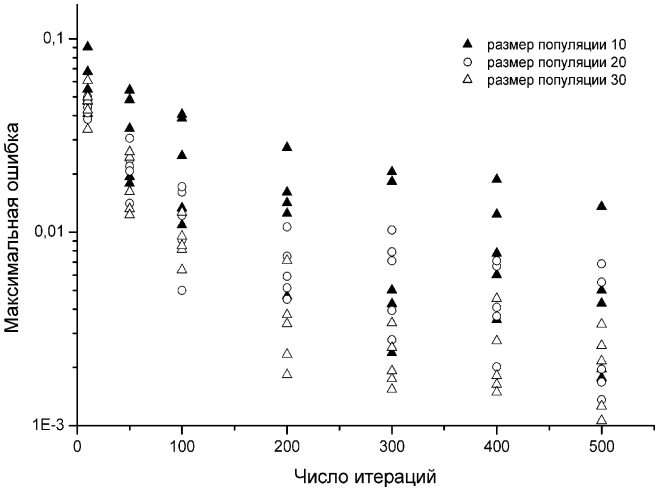


Рис. 3. Распределение максимальной ошибки решения

4,092857719066E-10	-1,5572501318133 3	-1,2693379959037 7	-0,7829620265420 53	-1,3898605072728 9	-0,6475591875962 19	0,0553394728039 57	…	-0,0034529249438 36	0,8136788438483 2	1,5644201604230 5	0,8095463569347 92	1,5257512931985 1	1,6043813039 98

Рис. 4. Фрагмент оптимального решения

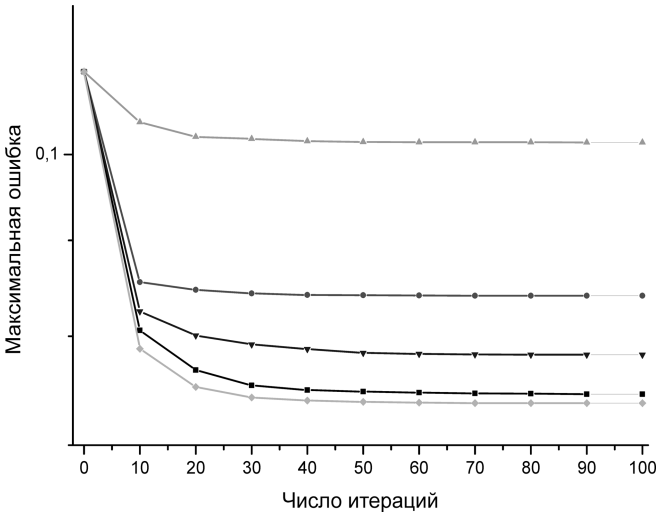


Рис. 5. Динамика поиска решений при $w = 0{,}7$; $c_1 = c_2 = 0{,}3$

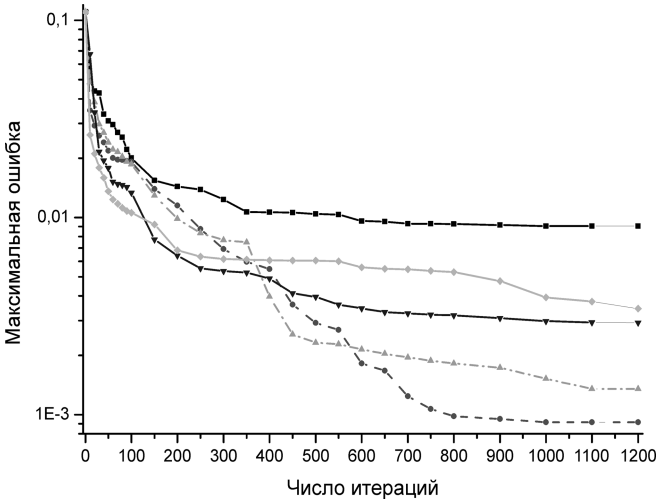


Рис. 6. Динамика поиска решений при $w = 0{,}7$; $c_1 = 0$; $c_2 = 1{,}4$

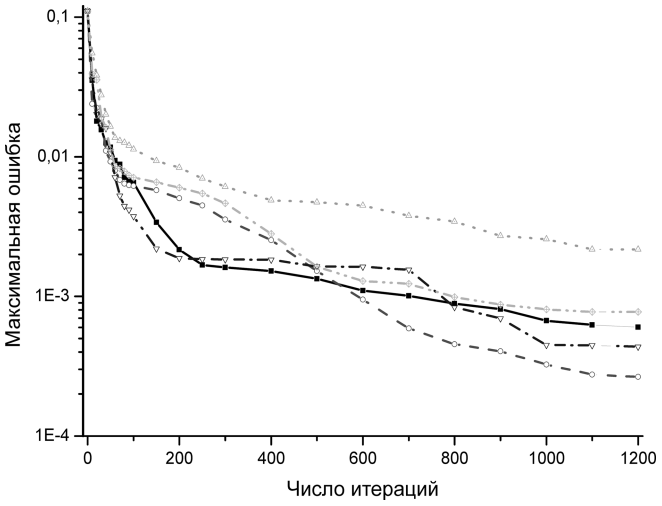


Рис. 7. Динамика поиска решений при $w = 0{,}7$; $c_1 = 1{,}4$; $c_2 = 1{,}4$

шего решения. В отдельных случаях при таком сочетании параметров ошибка может быть уменьшена на два порядка, этот факт иллюстрирует рис. 6.

При $c_1 > 0$ и $c_2 > 0$ на направление движения частиц влияют лучшая локальная позиция $\boldsymbol{p}_i(k)$ и лучшая позиция в рое $\boldsymbol{p}_g(k)$. Проведенные имитационные эксперименты позволили определить лучшие параметры для аппроксимации данной функции: $w = 0{,}7$; $c_1 = 1{,}4$; $c_2 = 1{,}4$, пять траекторий поиска решения приведены на рис. 7.

Коэффициент инерции больше 1. При $w > 1{,}1$ для поведения роя характерны большие скорости и, как следствие, большой разлет частиц и отсутствие сходимости алгоритма. После 500 итераций отдельные частицы задавали худшее решение с ошибкой, превышающей три единицы (лучшее найденное решение на этой итерации имело ошибку 0,0287366), и значения ошибок худших решений не уменьшались к 1000-й итерации. Таким образом, при коэффициенте инерции больше единицы рой не способен ни обнаружить, ни исследовать перспективные области решений.

Заключение

Алгоритм роящихся частиц принадлежит к классу методов, основанных на популяции. К этому же классу относится и генетический алгоритм. Сравним эти два алгоритма:

- оба алгоритма работают с множеством потенциальных решений (популяцией), основываясь на принципе сотрудничества между индивидами популяции;
- алгоритм роящихся частиц прост в реализации и не требует больших вычислительных затрат; здесь отсутствуют специфические генетические операторы кроссовера и мутации;
- частицы меняют позицию в пространстве поиска, не производя новых частиц, в отличие от генерации особей в генетическом алгоритме.

Применение алгоритма роящихся частиц для идентификации параметров нечетких моделей, связанное с выбором параметров алгоритма, требует настройки на конкретную таблицу наблюдений. Экспериментальные исследования позволили определить лучшие параметры для аппроксимации гладких функций.

Список литературы

1. **Espinosa J. Vandewalle J., Wertz. V.** Fuzzy logic, identification and predictive control. London: Springer-Verlag, 2005. 263 p.

2. **Dreo J., Petrowski A., Siarry P., Taillard E.** Metaheuristics for hard optimization. Methods and case studies. Berlin: Springer, 2006. 369 p.

3. **Kennedy J., Ebenhart R.** Particle Swarm Optimization // Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth: IEEE Service Center. 1995. P. 1942—1948.

4. **Parsopoulos K. E., Vrahatis M. N.** Recent approaches to global optimization problems through Particle Swarm Optimization // Natural Computing. 2002. V. 1. P. 235—306.

УДК 534.222.2

Ю. И. Димитриенко,
д-р физ.-мат. наук, проф., зав. каф.,
А. А. Захаров, аспирант,
МГТУ им. Н. Э. Баумана

Автоматизированная система для моделирования газовых потоков методом ленточных адаптивных сеток

Предложен метод построения автоматизированной системы для моделирования течений многомерных нестационарных газовых потоков в областях сложной формы. Метод основан на использовании адаптивных регулярных сеток. В качестве примера представлены результаты моделирования газовых потоков в каналах воздухозаборников сверхзвуковых летательных аппаратов.

Ключевые слова: *вычислительная газодинамика, криволинейные адаптивные сетки, автоматизированные системы, сверхзвуковые воздухозаборники.*

Введение

Численное моделирование газодинамических потоков является одним из важнейших элементов современного проектирования изделий в энергетическом машиностроении, авиации, ракетостроении, судостроении и многих других областях техники. В настоящее время для численного моделирования в газовой динамике применяются как коммерческие программные продукты (например, STAR-CD, FLUENT, FLOTRAN, FlowVision, ANSYS, GasDynamicsTool), так и собственные разработки компаний, занимающихся проектированием изделий, а также разработки университетов и научно-исследовательских институтов. Основная причина, почему коммерческие пакеты не вытеснили собственные разработки компаний, по-видимому, заключается в том, что каждый из коммерческих продуктов основан на использовании какого-либо одного из многочисленных методов вычислений, существующих в газовой динамике. Однако хорошо известно, что ни один из вычислительных методов не имеет абсолютных преимуществ по качеству получаемого решения и не является универсальным, пригодным для всего широкого набора задач газовой динамики. В последнее время стали активно развиваться так называемые методы адаптивных сеток, в которых важнейший элемент всех численных методов — разностная сетка — выбирается согласованно с границами геометрической области решения задачи газовой динамики, а иногда даже варьируется в ходе решения, изменяя свою "разрешающую способность" в зонах, где параметры газового потока меняются наиболее интенсивно. Коммерческие программные продукты с методами адаптивных сеток пока не известны, научные же исследования с их помощью ведутся весьма активно. В работах [1—4], выполненных в МГТУ им. Н. Э. Баумана на кафедре "Вычислительная математика и математическая физика", был разработан метод ленточных адаптивных сеток (ЛАС), который показал свою эффективность для расчета газовых потоков в изделиях сложной геометрической формы. В настоящей работе представлены результаты разработки автоматизированной системы моделирования газовых потоков методом ЛАС, в которой имеются, как и в каждом коммерческом продукте, все основные элементы для автоматизации вычислений: модуль 3D геометрического моделирования для задания облика конструкций, генератор разностной сетки, модуль задания свойств, нагрузок и начальных данных (препроцессор), решатель (процессор), а также модуль визуализации расчетов и постпроцессорной обработки данных.

Математическая постановка задачи газовой динамики

Разработанный метод ЛАС предназначен для решения общей 3-мерной системы уравнений динамики идеального газа, которая в криволинейных координатах X^i записывается следующим образом:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{W}^i}{\partial X^i} = 0;$$

$$\boldsymbol{U} = \sqrt{G}\begin{pmatrix} \rho \\ \rho Q_j^1 v^j \\ \rho Q_j^2 v^j \\ \rho Q_j^3 v^j \\ \rho E \end{pmatrix};\ \boldsymbol{W}^i = \sqrt{G}\begin{pmatrix} \rho v^i \\ Q_j^1(\rho v^i v^j + pG^{ij}) \\ Q_j^2(\rho v^i v^j + pG^{ij}) \\ Q_j^3(\rho v^i v^j + pG^{ij}) \\ v^i(\rho E + p) \end{pmatrix}, \quad (1)$$

где $\boldsymbol{U}$, $\boldsymbol{W}$ — координатные столбцы (комплексы); p — плотность газа; E — полная энергия газа,

$E = c_V\theta + \frac{|\boldsymbol{v}|^2}{2}$, c_V — теплоемкость при постоянном объеме, θ — температура газа, $\boldsymbol{v}$ — вектор скорости, $|\boldsymbol{v}|^2 = \boldsymbol{v}\cdot\boldsymbol{v}$ — квадрат модуля скорости; p — давление; $p = \rho R\theta$, R — газовая постоянная; $Q_i^j = \frac{\partial x^j}{\partial X^i}$ — якобиева матрица преобразования декартовых координат x^j в криволинейные X^i; $G_{ij} = Q_i^k Q_j^l \delta_{kl}$ — метрическая матрица, $G = \det G_{ij}$ — детерминант, v^j — компоненты вектора скорости в системе координат X^i.

Рассмотрим четыре наиболее часто встречающихся случая граничных условий для системы уравнений (1). Для удобства положим, что граница Σ области V состоит из четырех частей Σ_α, $\alpha = 1, ..., 4$, на каждой из которых заданы граничные условия соответствующего типа.

1. На границе Σ_1 с твердой поверхностью задается условие непротекания: $v_n = 0$, где $v_n = v^i n_i$, а n_i — компоненты вектора внешней нормали $\boldsymbol{n}$ к границе Σ_1.

2. На границе Σ_2, которая представляет собой искусственную границу сверхзвукового входа потока (на ней выполняются условия $v_n \leqslant 0$, $|\boldsymbol{v}| > M$, M — число Маха), задаются условия: $\rho = \rho_e$, $v^i = v_e^i$, $\theta = \theta_e$, где ρ_e, v_e^i, θ_e — заданные значения.

На входной границе Σ_2, на которой выполняются условия дозвукового входа потока $v_n \leqslant 0$, $|\boldsymbol{v}| < M$, задаются следующие условия: $\rho = \rho_e$, $v^i = v_e^i$.

3. На искусственной границе Σ_3, для которой выполняются условия дозвукового выхода потока $v_n \geqslant 0$, $|\boldsymbol{v}| < M$, задается одно условие, например для давления или плотности: $\rho = \rho_e$.

На выходной границе Σ_3 сверхзвукового выхода потока, на которой выполняются условия $v_n \geqslant 0$, $|\boldsymbol{v}| > M$, не задается никаких граничных условий [5—7].

4. На границе Σ_4, которая представляет собой плоскость симметрии области V, могут быть заданы условия симметрии:

$$n^i \frac{\partial \rho}{\partial X^i} = 0;\ v^i n_i = 0;\ n^i \frac{\partial v_{\tau_I}}{\partial X^i} = 0;\ n^i \frac{\partial \theta}{\partial X^i} = 0.$$

Начальные условия к системе (1) имеют вид:

$$t = 0:\ \rho(0, X^i) = \rho_0,\ v^i(0, X^j) = v_0^i,$$
$$E(0, X^i) = c_V\theta_0,$$

где ρ_0, θ_0, v_0^i — заданные значения.

Генерация регулярной адаптивной сетки в методе ЛАС

Рассматриваемую область V в пространстве R^3 адаптивных координат разобьем на совокупность криволинейных блоков V_m, каждый из которых имеет вид, изображенный на рис. 1 (рассматриваются только области, для которых это разбиение возможно). Для генерации адаптивной разностной сетки построим функции $x^i = f^i(X^j)$, преобразующие криволинейный блок V_m в декартовых координатах x^i в единичный куб $C = [0,1] \times [0,1] \times [0,1]$ в криволинейных (адаптивных) координатах X^j. Зададим стороны a, b, c, d, e, f (рис. 1) криволинейного блока V_m в параметрическом виде:

$$x^i = x_a^i(X^1, X^2),\ x^i = x_b^i(X^1, X^3),\ x^i = x_c^i(X^1, X^2),$$
$$x^i = x_d^i(X^1, X^3),\ x^i = x_e^i(X^2, X^3),\ x^i = x_f^i(X^2, X^3). \quad (2)$$

Искомое преобразование координат имеет вид

$$f^i(X^1, X^2, X^3) = P^i(X^1, X^2, X^3) - [P^i(0, X^2, X^3) - x_f^i(X^2, X^3)](1 - X^1) - X^1[P^i(1, X^2, X^3) - x_e^i(X^2, X^3)];$$

$$P^i(X^1, X^2, X^3) = T^i(X^1, X^2, X^3) - [T^i(X^1, 0, X^3) - x_d^i(X^1, X^3)](1 - X^2) - X^2[T^i(X^1, 1, X^3) - x_b^i(X^1, X^3)]; \quad (3)$$

$$T^i(X^1, X^2, X^3) = (1 - X^3)x_a^i(X^1, X^2) + X^3 x_c^i(X^1, X^2).$$

Если теперь ввести в координатах X^j для куба $C = [0,1] \times [0,1] \times [0,1]$ регулярную сетку

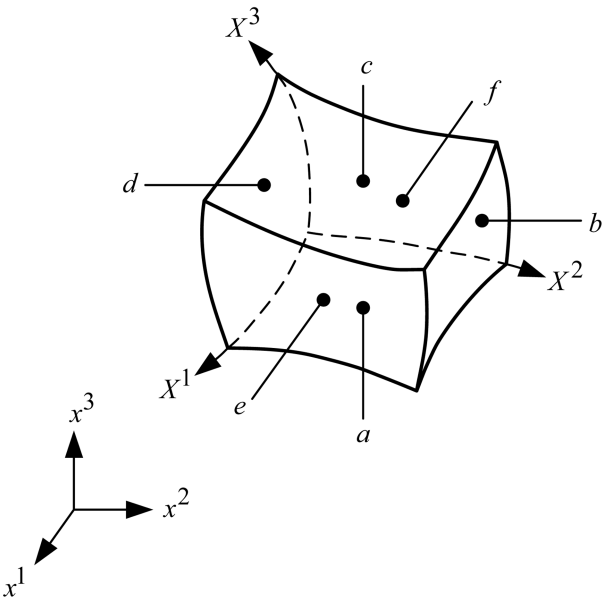


Рис. 1. Криволинейный блок и адаптивные координаты в нем

$C_m = \{X_i^1,\ X_j^2,\ X_k^3;\ i = 1...N,\ j = 1...M,\ k = 1...L\}$, то преобразование (3) этой сетки образует адаптивную сетку в координатах x^i: $V_m = \{x_i^1,\ x_j^2,\ x_k^3;\ i = 1...N,\ j = 1...M,\ k = 1...L\}$.

В основу метода компьютерной генерации криволинейной в x^i области V положен "обратный способ", когда в координатах X^j задается образ этой области Ξ как совокупность кубов. Если задать граничные функции (2) для каждого из кубов C_m, $m = 1, ..., \Xi$, то согласно формулам (3) будет определено преобразование области Ξ в координаты x^i и, тем самым, будет решена задача компьютерной генерации криволинейной области V.

Конечно-разностная аппроксимация уравнений газовой динамики в методе ЛАС

В методе ЛАС вместо традиционного трехиндексного перечисления узлов адаптивной сетки C_m $(X_i^1,\ X_j^2,\ X_k^3)$ используется одноиндексный способ, при котором все узлы сетки нумеруются одним индексом $(X_j^1,\ X_j^2,\ X_j^3)$. Такой способ дает возможность легко перечислить все узлы адаптивной сетки для области V, составленной из произвольного числа блоков V_m, — для нее индекс j просто принимает значения $j = 0, ..., K$, где K — общее число узлов (этим обосновывается название ленточной сетки). Заметим, что такой способ перечисления узлов сетки широко применяется в конечно-элементных методах и называется глобальной нумерацией узлов.

Отличительной особенностью разработанного алгоритма является то, что одновременно с генерацией сетки формируется список соседних узлов — по шесть для каждого j-го узла, которым присваиваются имена: B_j, F_j, L_j, R_j, D_j, U_j, обозначающие локальное расположение соседей j-го узла: "сзади (Behind)", "спереди (Front)", "слева (Left)", "справа (Right)", "снизу (Down)" и "сверху (Up)" соответственно (рис. 2).

Если имеется некоторая функция $h(X^1, X^2, X^3)$, то ее значение в узле $(X_j^1,\ X_j^2,\ X_j^3)$ при одноиндексном способе перечисления узлов обозначается как $h_j = h(X_j^1,\ X_j^2,\ X_j^3)$. Разностные аппроксимации производных при таком способе получают следующее обозначение (например, правая разность):

$$\frac{\partial h(X_j^1, X_j^2, X_j^3)}{\partial X_j^1} \approx \frac{h_{F_j} - h_j}{X_{F_j}^1 - X_j^1}.$$

Используя этот алгоритм аппроксимации частных производных, можно строить различные аппроксимации системы уравнений газовой динамики (1), выбирая при этом тот или иной конечно-разностный метод. Покажем применение алгоритма для одного из наиболее простых по алгоритмизации конечно-разностных методов решения задач газовой динамики — метода, аналогичного методу Мак—Кормака, который широко распространен для решения задач газовой динамики в ортогональных координатах.

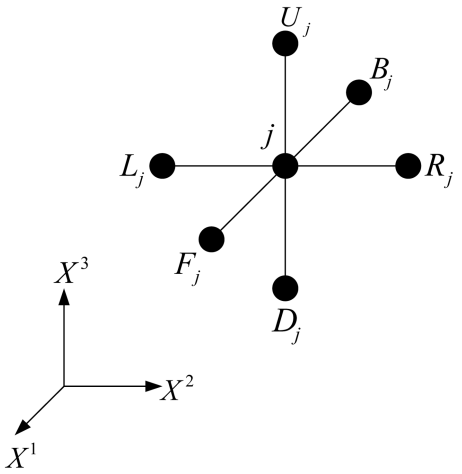


Рис. 2. Узел разностной сетки и его 6 соседей в 3-мерной ЛАС

Метод состоит из трех шагов: предиктора, корректора и шага искусственной вязкости, который вводится для устранения эффекта нефизических осцилляций.

Шаг 1 (*предиктор*):

$$\boldsymbol{U}_j^{m+1/2} = \boldsymbol{U}_j^m - \Delta t \left(\frac{\boldsymbol{W}_{F_j}^{1,m} - \boldsymbol{W}_j^{1,m}}{X_{F_j}^1 - X_j^1} + \frac{\boldsymbol{W}_{R_j}^{2,m} - \boldsymbol{W}_j^{2,m}}{X_{R_j}^2 - X_j^2} + \frac{\boldsymbol{W}_{U_j}^{3,m} - \boldsymbol{W}_j^{3,m}}{X_{U_j}^3 - X_j^3} \right).$$

Шаг 2 (*корректор*).

$$\tilde{\boldsymbol{U}}_j^{m+1} = \frac{1}{2}(\boldsymbol{U}_j^m + \boldsymbol{U}_j^{m+1/2}) - \frac{\Delta t}{2} \left(\frac{\boldsymbol{W}_j^{1,m+1/2} - \boldsymbol{W}_{B_j}^{1,m+1/2}}{X_j^1 - X_{B_j}^1} + \frac{\boldsymbol{W}_j^{2,m+1/2} - \boldsymbol{W}_{L_j}^{2,m+1/2}}{X_j^2 - X_{L_j}^2} + \frac{\boldsymbol{W}_j^{3,m+1/2} - \boldsymbol{W}_{D_j}^{3,m+1/2}}{X_j^3 - X_{D_j}^3} \right).$$

Здесь обозначены:

$$\boldsymbol{U}_j^m = \boldsymbol{U}(t^m,\ X_j^1,\ X_j^2,\ X_j^3);$$

$$\boldsymbol{W}_j^{1,m} = \boldsymbol{W}^1(\boldsymbol{U}_j^m),\ \boldsymbol{W}_j^{2,m} = \boldsymbol{W}^2(\boldsymbol{U}_j^m),\ \boldsymbol{W}_j^{3,m} = \boldsymbol{W}^3(\boldsymbol{U}_j^m);$$

$$\boldsymbol{W}_j^{1,m+1/2} = \boldsymbol{W}^1(\boldsymbol{U}_j^{m+1/2}),\ \boldsymbol{W}_j^{2,m+1/2} = \boldsymbol{W}^2(\boldsymbol{U}_j^{m+1/2}),\ \boldsymbol{W}_j^{3,m+1/2} = \boldsymbol{W}^3(\boldsymbol{U}_j^{m+1/2}).$$

Шаг 3 (*искусственная вязкость*):

$$U_j^{m+1} = \tilde{U}_j^{m+1} + \alpha L(U_j^m),$$

где $L(U_j^m) = U_{F_j}^m + U_{R_j}^m + U_{U_j}^m - 6U_j^m + U_{B_j}^m + U_{L_j}^m + U_{D_j}^m$.

Параметр α — коэффициент вязкости.

Представленная разностная схема обеспечивает второй порядок точности аппроксимации. Чтобы сохранить этот порядок и для граничных узлов, введем так называемые фиктивные узлы, принадлежащие границе области $\overline{V}$, отличающейся от области V дополнительным слоем ячеек сетки, "оконтуривающим" область V.

1. Параметры на границе Σ_1 в методе фиктивных ячеек аппроксимируются следующим образом:

$$v_{nj} = -v_{nj_Z},\ \rho_j = \rho_{j_Z},\ v_{\tau j} = v_{\tau j_Z},\ \theta_j = \theta_{j_Z},$$

где v_{nj} — значения функций в граничном "фиктивном" узле j, а v_{nj_Z} — значения функций в узле j_Z, который получен с помощью двукратного применения оператора A, т. е. это "внутренний" узел для узла j_A, "внутреннего" к j: $j_Z = (j_A)_A$. Оператор A формирования номера граничного узла определяется в соответствии с правилом обозначения узлов (см. рис. 2) следующим образом:

$$A = \begin{cases} F_j, \text{ если узел } j \in \text{границе } X^1 = \text{const и } \Delta X^1 > 0, \\ B_j, \text{ если узел } j \in \text{границе } X^1 = \text{const и } \Delta X^1 < 0, \\ R_j, \text{ если узел } j \in \text{границе } X^2 = \text{const и } \Delta X^2 > 0, \\ L_j, \text{ если узел } j \in \text{границе } X^2 = \text{const и } \Delta X^2 < 0, \\ U_j, \text{ если узел } j \in \text{границе } X^3 = \text{const и } \Delta X^3 > 0, \\ D_j, \text{ если узел } j \in \text{границе } X^3 = \text{const и } \Delta X^3 < 0, \end{cases}$$

где ΔX^i — приращение координаты в направлении внутренности области V.

2. Параметры на границе Σ_2 сверхзвукового входа потока аппроксимируются следующим образом:

$$\rho_j = 2\rho_e - \rho_{j_Z},\ \ v_j^I = 2v_e^I - v_{j_Z}^I,$$
$$\theta_j = 2\theta_e - \theta_{j_Z}.$$

На дозвуковой границе входа потока значения функций аппроксимируются так:

$$\rho_j = 2\rho_e - \rho_{j_Z},\ v_j^I = 2v_e^I - v_{j_Z}^I,\ \theta_j = \theta_{j_Z}.$$

3. Значения на границе Σ_3 дозвукового выхода потока аппроксимируются следующим образом:

$$\rho_j = 2\rho_e - \rho_{j_Z},\ v_j^I = v_{j_Z}^I,\ \theta_j = \theta_{j_Z},$$

а на сверхзвуковой границе — так:

$$\rho_j = \rho_{j_Z},\ v_j^I = v_{j_Z}^I,\ \theta_j = \theta_{j_Z}.$$

4. Граничные значения для функций на поверхности симметрии Σ_4 аппроксимируются следующим образом:

$$n_{j_A}^1 \frac{(h_{(j_A)_F} - h_{(j_A)_B})}{(X_{(j_A)_F}^1 - X_{(j_A)_B}^1)} + n_{j_A}^2 \frac{(h_{(j_A)_R} - h_{(j_A)_L})}{(X_{(j_A)_R}^2 - X_{(j_A)_L}^2)} + n_{j_A}^3 \frac{(h_{(j_A)_U} - h_{(j_A)_D})}{(X_{(j_A)_U}^3 - X_{(j_A)_D}^3)} = 0,$$

$$h = \{\rho, \theta, v_\tau\},\ v_{nj} = -v_{nj_Z},$$

здесь индекс $(j_A)_R$ означает, что берется "правый" узел для j_A, индекс $(j_A)_L$ соответствует "левому" узлу для j_A и т. д.

Разработка пакета прикладных программ для автоматизированного моделирования газодинамических потоков

На основе описанного выше метода был разработан пакет прикладных программ, реализующий все основные стадии моделирования — препроцессор, процессор и постпроцессор. Препроцессор позволяет сначала создавать прямоугольные области из блоков-примитивов, а потом модифицировать их, превращая в криволинейные. Модификация области происходит с помощью перемещения узловых точек сплайнов (бисплайнов) на ее границе. Перемещение осуществляется либо мышью, либо с помощью непосредственного ввода координат узловой точки. Поддерживается также режим редактирования координат для выделенной группы точек. Доступны операции выделения границ и областей для задания на них граничных и начальных условий, настройки и запуска процесса генерации сетки. Сгенерированная сетка подается на вход процессору, осуществляющему непосредственное решение системы уравнений (1) и выдачу рассчитанных параметров в узлах сетки построцессору. Постпроцессорный модуль визуализирует рассчитанные сеточные данные в виде цветовых фонов.

Моделирование газовых потоков в воздухозаборниках СПВРД

Разработанный пакет прикладных программ применялся для моделирования газовых потоков в воздухозаборнике (ВЗ) модельного сверхзвукового прямоточного воздушно-реактивного двигателя (СПВРД). Были проведены расчеты для двух разных конфигураций ВЗ: с пилонами (конструктивный элемент, соединяющий внешнюю оболочку ВЗ с центральным телом) и без пилонов (рис. 3, см. третью сторону обложки).

Задача рассматривалась в трехмерной постановке. Результаты приведены для установившегося режима. Значения параметров представлены в безразмерном виде, в качестве характерных значений приняты следующие значения величин:

$$x_0 = 0{,}1 \text{ м};\ \theta_0 = 293 \text{ К};\ \rho_0 = 1{,}2928 \frac{\text{кг}}{\text{м}^3};$$

$$v_0 = \sqrt{\gamma R \theta}\ \frac{\text{м}}{\text{с}};\ \gamma = 1{,}4;\ R = 286{,}7 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}.$$

Параметры набегающего потока:

$$\rho_e = 0{,}1946 \frac{\text{кг}}{\text{м}^3};\ \boldsymbol{v}_e = v_{ze}\boldsymbol{e}_z;\ v_{ze} = 900 \frac{\text{м}}{\text{с}};\ \theta_e = 216{,}6 \text{ К}.$$

Начальные значения: $\rho_0 = \rho_e$; $\boldsymbol{v}_0 = 0{,}1\boldsymbol{v}_e$; $\theta_0 = \theta_e$.

В целях выяснения особенностей течения на входе в канал ВЗ было проведено дополнительное моделирование в рамках осесимметричной постановки задачи газовой динамики на существенно более мелкой сетке — с числом ячеек порядка 27 000. Представленные на рис. 4 (см. четвертую сторону обложки) данные позволяют говорить о достаточно высоком качестве численного моделирования, достигаемого методом ЛАС. Метод позволяет установить картину сложного движения газа на входе в канал ВЗ, внутри канала ВЗ, а также на внешней обтекаемой поверхности ВЗ с образованием нескольких скачков уплотнения.

Представленные результаты показывают, что после перехода на более мелкую сетку происходит существенное уменьшение зоны размазывания скачков и "проявляются" косые скачки в канале ВЗ, вызванные отражением волны от стенок канала, появление которых предсказывает теория скачков в ВЗ СПВРД [6, 7].

Оценивая суммарное машинное время расчета по всем конфигурациям (см. таблицу), можно сделать вывод о том, что многомерные нестационарные задачи являются достаточно трудоемкими для современных персональных компьютеров с частотой процессоров около 1,5...2,0 ГГц. Минимальное время счета до установления даже на грубой сетке составляло не менее 3—4 дней для трехмерных задач, что примерно на порядок больше времени расчета эквивалентных двумерных задач. Минимальное время счета до установления в осесимметричной задаче с достаточно мелкой сеткой составило порядка суток, что хотя и меньше времени решения трехмерных мерных задач, но все же весьма значительно. Расчет сложных конфигураций ВЗ такого типа становится более эффективным с применением технологий параллельных вычислений [1].

Данные об использованной разностной сетке и численных результатах решения трехмерной задачи для разных конфигураций ВЗ

Параметры расчета	Конфигурация без пилонов	Конфигурация с пилонами	Осесимметричный расчет
Число узлов сетки	2 240	4 992	27 000
Число итераций до установления	500 000	700 000	710 000
Время счета (ч) (процессор с частотой 2 ГГц)	72	178	21

Заключение

Разработанный метод ленточно-адаптивных регулярных сеток и пакет прикладных программ, реализующий этот метод, позволяют осуществлять моделирование многомерных нестационарных газодинамических процессов в областях со сложной геометрией. На основе этого метода проведено численное моделирование газодинамических процессов в канале воздухозаборника типового сверхзвукового прямоточного воздушно-реактивного двигателя, с помощью которого установлено, что для получения качественного решения с характерными для данного типа конструкций скачками уплотнения необходимо применять достаточно мелкие конечно-разностные сетки, содержащие, по крайней мере, несколько десятков узлов на один характерный линейный размер рассматриваемой области.

В ходе трехмерного моделирования было также установлено, что пилоны ВЗ существенно влияют на осевую скорость, давление, плотность и температуру потока, что говорит о необходимости их учета при газодинамических расчетах конструкций ВЗ перспективных СПВРД.

Список литературы

1. **Димитриенко Ю. И., Ануфриев С. Н., Изотова С. Г.** Разработка метода решения трехмерной нестационарной внутренней задачи газовой динамики на многопроцессорных вычислительных системах // Математика в современном мире / Под ред. Ю. А. Дробышева. Калуга: Изд-во КГПУ, 2004. С. 139—146.

2. **Димитриенко Ю. И., Изотова С. Г., Ануфриев С. Н., Захаров А. А.** Численное моделирование трехмерных газодинамических процессов в камерах сгорания РДДТ на основе метода геометрически-адаптивных сеток // Вестник МГТУ им. Н. Э. Баумана. Сер. Естественные науки. 2005. № 3. С. 45—58.

3. **Димитриенко Ю. И., Кукленков Л. Л., Ануфриев С. Н.** Метод ленточных адаптивных сеток для решения задач газовой динамики в невыпуклых областях сложной формы // Современные естественно-научные и гуманитарные проблемы: Сб. трудов научно-методической конференции, посвященной 40-летию НУК ФН. — М.: Логос, 2005. С. 506—512.

4. **Димитриенко Ю. И., Захаров А. А.** Разработка метода ленточно-адаптивных сеток для решения трехмерных задач газовой динамики в воздухозаборниках // Вестник МГТУ им. Н. Э. Баумана. Сер. Естественные науки. 2006. № 3. С. 44—56.

5. **Самарский А. А., Попов Ю. П.** Разностные методы решения задач газовой динамики. М.: Едиториал УРСС, 2004. 424 с.

6. **Численное** решение многомерных задач газовой динамики / Ред. С. К. Годунов. М.: Наука, 1976. 400 с.

7. **Гильманов А. Н.** Методы адаптивных сеток в задачах газовой динамики. М.: Наука, 2000. 248 с.

УДК 004.3

Ш. А. Оцоков, канд. техн. наук, доц.,
Московский энергетический институт
(технический университет)

Обобщение вычислений над полем комплексных чисел с исключением ошибок округления

Предложена обобщенная арифметика с исключением ошибок округления Грегори—Кришнамурти над полем комплексных чисел с целой действительной и мнимой частью, которая позволяет свести арифметические операции с комплексными числами к соответствующим операциям с целыми числами в модулярной системе счисления. Получены оценки модуля, необходимого для представления комплексных дробей Фарея.

Ключевые слова: *комплексная дробь Фарея, фареевский квадрат, целое комплексное Фарея.*

Существующие плохо обусловленные вычислительные задачи можно условно разбить на следующие категории:

- *плохо обусловленные задачи с неточно заданными исходными данными*;
- *плохо обусловленные задачи с точно заданными исходными данными*;
- *вычислительные задачи, которые становятся плохо обусловленными при некоторых условиях.*

Известно, что характерное свойство плохо обусловленных вычислительных задач — это высокая чувствительность к изменениям исходных данных и точности компьютерных вычислений [1, 3]. Задачи первой категории бессмысленно решать путем увеличения точности компьютерных вычислений, так как даже если все арифметические операции будут выполнены абсолютно точно, решение все равно не будет верным [1]. Что касается второй и третьей категорий, то здесь точность компьютерных вычислений может играть решающую роль.

Известным примером задачи второй категории является обращение матрицы Гильберта [2]. К задаче третьей категории можно отнести, например, двумерную краевую задачу для дифференциальных уравнений второго порядка. Так, при численном решении этой задачи методом конечных элементов с применением равномерной сетки с линейными функциями формы имеет место зависимость [4]

$$\alpha = Ch^{-2}, \qquad (1)$$

где α — спектральное число обусловленности матрицы системы алгебраических уравнений; C — константа, зависящая от задачи; h — максимальный размер элемента.

Из (1) видно, что с измельчением конечно-элементной сетки ($h \to 0$) ухудшается обусловленность матрицы разрешающих алгебраических уравнений, и даже малые ошибки округления при численном решении задачи могут привести к решению, сильно отличающемуся от искомого точного [4].

Точность вычислений с плавающей точкой, поддерживаемая современными сопроцессорами, в ряде случаев может быть недостаточной для численного решения плохо обусловленных задач второй и третьей категорий. Это привело к исследованию нетрадиционных числовых систем (модулярной, p-адической) и арифметик, которые исключают ошибки округления над полем рациональных чисел [3].

Известно, что любое целое число в модулярной системе (МС) представляется в виде остатка от деления на выбранный модуль M (простое число), арифметические операции сложения, вычитания, умножения в МС проводятся над остатками. Если результат этих операций положительный и больше M, то находится остаток от деления результата на этот модуль; если отрицательный — дополнение до модуля. Например, $M = 17$, A $= 20$, $B = 13$, тогда

$$A + B = 33;$$

$$A - B = 7;$$

$$A \cdot B = 260,$$

в МС получим:

$$A \bmod M \equiv 20 \bmod 17 \equiv 3,$$

$$B \bmod M \equiv 13 \bmod 17 \equiv 1\ 3,$$

$$(A + B) \bmod M \equiv 16 \bmod 17 \equiv 16,$$

$$(A - B) \bmod M \equiv -10 \bmod 17 \equiv 7,$$

$$(A \cdot B) \bmod M \equiv 39 \bmod 17 \equiv 5.$$

Так как операция деления заменяется на операцию умножения делимого на мультипликативно обратный элемент делителя, то результатом операции над двумя целыми числами является целое число, например:

$$(A/B) \bmod M \equiv (A \cdot B^{-1}) \bmod M \equiv$$
$$\equiv (3 \cdot 4) \bmod M \equiv 12.$$

Таким образом, в МС все арифметические операции проводятся над целыми числами, и результатом любой такой операции опять является целое число, что исключает ошибки округления.

Модель вычислений с исключением ошибок округления над полем рациональных чисел на основе одномодульной арифметики представлена на рис. 1.

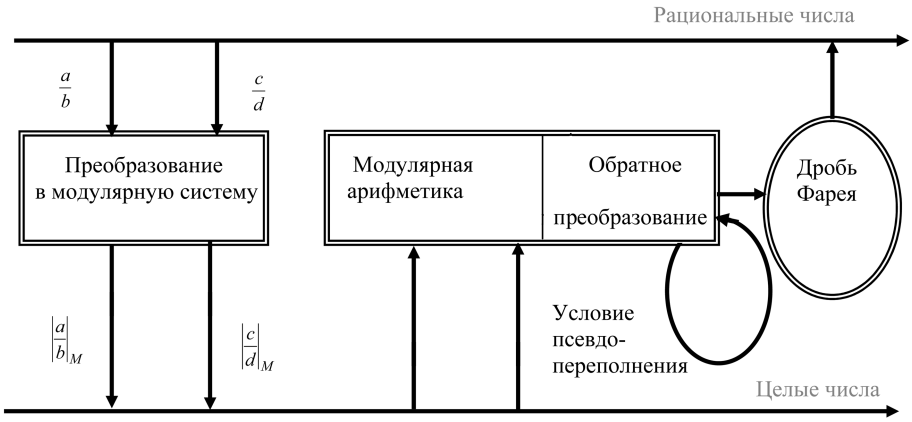


Рис. 1. Модель последовательных вычислений над полем рациональных чисел с исключением ошибок округления

В соответствии с этой моделью исходные данные — дроби $\frac{a}{b}$, $\frac{c}{d}$ — преобразуются в МС и представляются в ней в виде целых чисел $\left|\frac{a}{b}\right|_M$, $\left|\frac{c}{d}\right|_M$ (M — модуль). Далее с этими целыми числами проводятся все вычисления по правилам модулярной арифметики. Полученные результаты — целые числа — отображаются в несократимые дроби (дроби Фарея). Существует бесконечно много несократимых дробей, которые представляются в МС одним и тем же числом, и чтобы обеспечить единственность представления дробей в МС, в работе [3] были введены дроби Фарея (несократимые дроби, числители и знаменатели которых по абсолютной величине меньше некоторой константы — порядка дроби Фарея). Например, дроби Фарея порядка 2 имеют вид:

$$\frac{0}{1}, \frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{-1}{1}, \frac{-1}{2}, \frac{-2}{1}.$$

В [3] доказана теорема единственности, которая утверждает, что не существует двух или более дробей Фарея порядка N, которые представляются в МС одним и тем же числом, если выполняется неравенство

$$2N^2 + 1 \leqslant M, \qquad (2)$$

где M — модуль; N — порядок дробей.

В табл. 1 приведены дроби Фарея порядка 2 и соответствующие им целые числа по модулю $M = 7$.

Из таблицы видно, что операции над дробями Фарея 1/2 − 1 = -1/2 соответствует операция над целыми: 4 − 1 = 3. Рассмотрим еще один пример: 1/2 + 1 = 3/2, что соответствует операции над целыми: 4 + 1 = 5, но по таблице $5 \leftrightarrow -2$, т. е. получен неверный ответ в МС. Эта ситуация называется псевдопереполнением [3], т. е. порядок дроби Фарея искомого результата не удовлетворяет неравенству (2), в данном случае порядок искомого результата равен 3, и он не удовлетворяет неравенству (2).

Известно, что для численного решения некоторых задач в электротехнике, энергетике и других областях науки и техники возникает необходимость выполнения арифметических операций с комплексными числами. Как правило, арифметические операции с комплексными числами $z_1 = x_1 + iy_1$, $z_2 = x_2 + iy_2$ на ЭВМ выполняются по следующим формулам:

$$z_1 \pm z_2 = (x_1 \pm x_2) + i(y_1 \pm y_2);$$
$$z_1 \cdot z_2 = (x_1x_2 - y_1y_2) + i(x_1y_2 + y_1x_2); \qquad (3)$$
$$\frac{z_1}{z_2} = \frac{x_1x_2 + y_1y_2}{x_2^2 + y_2^2} + i\frac{y_1x_2 - x_1y_2}{x_2^2 + y_2^2}.$$

Так как вычисления по формулам (3) проводятся в арифметике с плавающей точкой, то в процессе выполнения вычислений с комплексными числами неизбежны ошибки округления. В табл. 2 приведены результаты решения плохо обусловленной задачи второй категории — обращения матрицы, коэффициенты которой определяются по формулам

$$h_{kz} = \frac{i}{k + z - 1},$$

где k, z — номера строки и столбца матрицы.

Цель настоящей работы состоит в обобщении алгоритмов одномодульной арифметики, которые представлены в работе [3], для реализации вычислений с исключением ошибок округления над полем рациональных комплексных чисел (рациональное комплексное число — это число, представимое в виде дроби, числитель и знаменатель которой — комплексные числа с целой вещественной и мнимой частями). Рассмотрим вначале частный случай — одномодульную арифметику целых ком-

Таблица 1

Дробь Фарея	0	1	1/2	2	−1	−1/2	−2
МС	0	1	4	2	6	3	5

Таблица 2

Порядок матрицы	Относительная погрешность последнего диагонального элемента обратной матрицы, %
5	$5 \cdot 10^{-11}$
8	$7 \cdot 10^{-7}$
10	>100

плексных чисел (целое комплексное число — это число с целой мнимой и вещественной частями).

Одномодульная арифметика целых комплексных чисел. Впервые понятие целого комплексного числа было введено Гауссом и им же была доказана следующая теорема [5].

Фундаментальная теорема Гаусса. По заданному комплексному модулю $m = p + iq$, норма которого равна $R = p^2 + q^2$ и для которого p, q являются взаимно простыми числами, каждое целое комплексное число сравнимо с одним и только одним вычетом из ряда

$$0, 1, 2, 3, \ldots, R - 1.$$

Справедлива ***теорема*** [5]. Если $A = a + bi$, $m = p + iq$ и выполняются сравнения

$$\begin{aligned} ap + bq &\equiv xp + yq (\mathrm{mod}\ p^2 + q^2); \\ bp - aq &\equiv yp - xq (\mathrm{mod}\ p^2 + q^2), \\ \text{то } A &\equiv x + iy (\mathrm{mod} M). \end{aligned} \qquad (4)$$

Таким образом, при выполнении условия (4) число $x + iq$ является вычетом числа A по модулю M.

Из (4) следует, что множество ***наименьших комплексных вычетов*** по модулю M определяется из системы неравенств [5]:

$$\begin{cases} 0 \leqslant xp + yq < p^2 + q^2, \\ 0 \leqslant yp - xq < p^2 + q^2 \end{cases}$$

или

$$\begin{cases} 0 \leqslant \dfrac{xp + yq}{p^2 + q^2} < 1. \\ 0 \leqslant \dfrac{yp - xq}{p^2 + q^2} < 1. \end{cases}$$

Множество ***абсолютно наименьших комплексных вычетов*** по модулю M определяется из системы неравенств [5]:

$$\begin{cases} -\dfrac{1}{2} \leqslant \dfrac{xp + yq}{p^2 + q^2} < \dfrac{1}{2}; \\ -\dfrac{1}{2} \leqslant \dfrac{yp - xq}{p^2 + q^2} \leqslant \dfrac{1}{2}. \end{cases} \qquad (5)$$

Система неравенств (5) определяет некоторое множество, и для его геометрической интерпретации необходимо решить систему линейных уравнений, которая определяет вершины этого множества [5]:

$$\begin{cases} yp - xq = -\dfrac{p^2 + q^2}{2} \quad (PN); \\ yp - xq = \dfrac{p^2 + q^2}{2} \quad (QR); \\ xp + yq = -\dfrac{p^2 + q^2}{2} \quad (RN); \\ xp + yq = \dfrac{p^2 + q^2}{2} \quad (PQ). \end{cases} \qquad (6)$$

На рис. 2 представлена геометрическая интерпретация множества абсолютно наименьших вычетов по модулю M.

Таким образом, точки, лежащие внутри фигуры $QPNR$ (рис. 2), являются абсолютно наименьшими вычетами по модулю M, а точки, лежащие вне фигуры $QPNR$, — целыми комплексными числами, не удовлетворяющими системе (5) и не представимыми по модулю M. В (6) после каждого уравнения в скобках указано уравнение соответствующей прямой (рис. 2).

Определим множество ***целых комплексных чисел*** Фарея порядка N следующим образом:

$$\begin{aligned} A = \{x + iy,\ x \in Z,\ y \in Z,\ 0 \leqslant |x| \leqslant N, \\ 0 \leqslant |y| \leqslant N\}, \end{aligned} \qquad (7)$$

где Z — множество целых чисел, а число $N > 0$ назовем *порядком комплексных целых чисел Фарея*. При $N = 3$ множество A будет состоять из следующих чисел:

$-3 - 3i$, $-3 - 2i$, $-3 - i$, -3, $-3 + i$, $-3 + 2i$, $-3 + 3i$,
$-2 - 3i$, $-2 - 2i$, $-2 - i$, -2, $-2 + i$, $-2 + 2i$, $-2 + 3i$,
$-1 - 3i$, $-1 - 2i$, $-1 - i$, -1, $-1 + i$, $-1 + 2i$, $-1 + 3i$,
$-3i$, $-2i$, $-i$, 0, i, $2i$, $3i$,
$1 - 3i$, $1 - 2i$, $1 - i$, 1, $1 + i$, $1 + 2i$, $1 + 3i$,
$2 - 3i$, $2 - 2i$, $2 - i$, 2, $2 + i$, $2 + 2i$, $2 + 3i$,
$3 - 3i$, $3 - 2i$, $3 - i$, 3, $3 + i$, $3 + 2i$, $3 + 3i$.

Геометрическая интерпретация множества целых комплексных чисел Фарея порядка N представлена на рис. 3. Назовем его ***фареевским квадратом.***

Из рис. 3 видно, что во множестве A будет $(2N + 1)^2$ целых комплексных чисел Фарея. Модуль M определяет множество представимых целых комплексных чисел, но возникает вопрос: каков максимальный порядок целых комплексных чисел Фарея N, представимых при заданном модуле M? Или тот же самый вопрос, но сформулированный иначе: какова максимальная длина стороны фареевского квадрата, вписывающегося в ромб, изо-

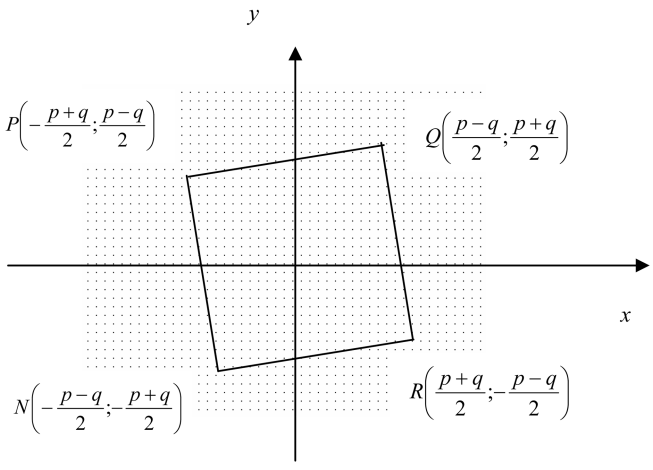


Рис. 2. Геометрическая интерпретация множества абсолютно наименьших вычетов

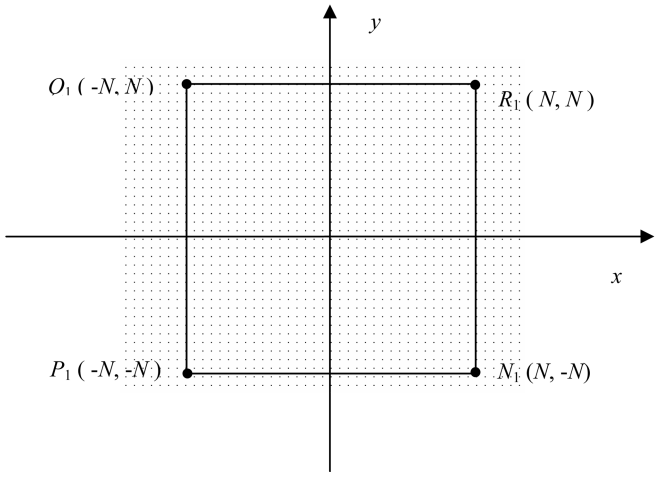


Рис. 3. Геометрическая интерпретация множества целых комплексных чисел Фарея порядка *N*

браженный на рис. 2? Справедлива следующая оценка:

$$N \leqslant \left[\frac{p}{2} - \frac{q}{2}\right], \qquad (8)$$

где $[x]$ — целая часть x и предполагается, что $p > q > 0$.

Доказательство следует из рис. 4.

Из формулы (8) следует, что при уменьшении мнимой части модуля увеличивается порядок представимых целых комплексных чисел Фарея.

Пример. Выберем модуль $M = 10 + i$.

Здесь и далее везде выбирать модуль $M = p + iq$ будем исходя из того, что $p^2 + q^2$ — простое число, а из фундаментальной теоремы Гаусса известно, что любое целое комплексное число сравнимо

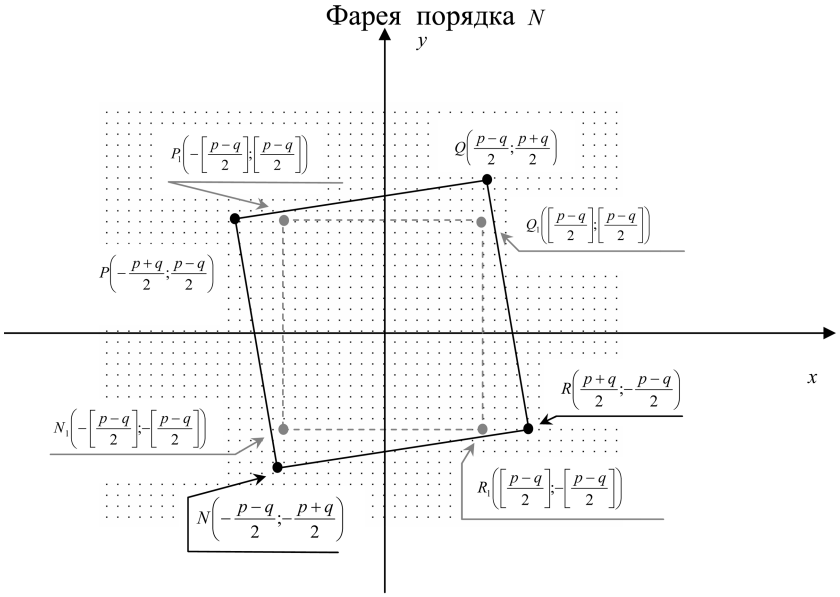


Рис. 4. Фареевский квадрат $P_1Q_1R_1N_1$, вписанный в ромб $PQRN$

только с одним целым числом по модулю $p^2 + q^2$. Из (8) следует, что порядок целого комплексного числа Фарея $N = 4$.

Ниже приведена таблица всех целых комплексных чисел Фарея и соответствующих им целых чисел по модулю $p^2 + q^2 = 101$ (табл. 3). Символом C_z в ней обозначены строки, содержащие целые комплексные числа Фарея, символом Z — соответствующие целые числа по модулю $p^2 + q^2 = 101$.

Найдем значение выражения D_1:

$$D_1 = (-1 + 3i) \cdot (-1 + i) + 2i - (2 - 4i).$$

После упрощений оно равно $(-4 + 2i)$.

Найдем значение этого выражения, пользуясь табл. 3:

$$D_1 = ((-31) \cdot (-11) + (-20) - (42)) \bmod 101 = 77.$$

Так как $77 > \frac{p^2 + q^2}{2}$, то $D_1 = -(101 - 77) = -24$,

из табл. 3 видно, что **(−24)** соответствует целому комплексному числу $(-4 + 2i)$.

Найдем значение выражения D_2:

$$D_2 = (-1 + 3i) + (-1 + 4i) = -2 + 7i.$$

Пользуясь табл. 3, получим

$$D_2 = ((-31) + (-41)) \bmod 101 = 29.$$

Из табл. 3 видно, что **29** соответствует целому комплексному числу $(-1 - 3i)$. Ответ неверный, возникла ошибка псевдопереполнения, так как истинный искомый результат $(-2 + 7i)$ есть целое комплексное число Фарея порядка 7, а не 4.

Алгоритмы преобразования вида {Целое комплексное → в МС и МС → Целое комплексное число} рассмотрены в работе [5]. На сайте [6] размещены исходные тексты программ на Delphi, в которых эти алгоритмы реализованы.

Рассмотрим общий случай — одномодульную арифметику рациональных комплексных чисел, т. е. таких чисел, которые представляются в виде отношения двух целых комплексных чисел.

Одномодульная арифметика рациональных комплексных чисел. По аналогии с (7) определим множество *комплексных дробей Фарея* порядка N следующим образом:

$$B = \begin{cases} \dfrac{z_1}{z_2} = \dfrac{x + iy}{a + ib},\ x \in Z,\ y \in Z, \\ a \in Z,\ b \in Z, \\ 0 \leqslant |x| \leqslant N,\ 0 \leqslant |y| \leqslant N, \\ 0 \leqslant |a| \leqslant N,\ 0 \leqslant |b| \leqslant N, \\ z_2 \neq 0,\ \text{НОД}\ (z_1, z_2) = 1, \end{cases} \qquad (9)$$

Таблица 3

Cz	$-5+0i$	$-5+1i$	$-5+2i$	$-5+3i$	$-5+4i$	$-4+-5i$	$-4+-4i$	$-4+-3i$	$-4+-2i$	
Z	-5	-15	-25	-35	-45	46	36	26	16	
Cz	$-4+-1i$	$-4+0i$	$-4+1i$	$-4+2i$	$-4+3i$	$-4+4i$	$-3+-5i$	$-3+-4i$	$-3+-3i$	$-3+-2i$
Z	6	-4	-14	**-24**	-34	-44	47	37	27	17
Cz	$-3+-1i$	$-3+0i$	$-3+1i$	$-3+2i$	$-3+3i$	$-3+4i$	$-2+-5i$	$-2+-4i$	$-2+-3i$	$-2+-2i$
Z	7	-3	-13	-23	-33	-43	48	38	28	18
Cz	$-2+-1i$	$-2+0i$	$-2+1i$	$-2+2i$	$-2+3i$	$-2+4i$	$-1+-5i$	$-1+-4i$	$-1+-3i$	$-1+-2i$
Z	8	-2	-12	-22	-32	-42	49	39	**29**	19
Cz	$-1+-1i$	$-1+0i$	$-1+1i$	$-1+2i$	$-1+3i$	$-1+4i$	$0+-5i$	$0+-4i$	$0+-3i$	$0+-2i$
Z	9	-1	**-11**	-21	**-31**	**-41**	50	40	30	20
Cz	$0+-1i$	$0+0i$	$0+1i$	$0+2i$	$0+3i$	$0+4i$	$0+5i$	$1+-4i$	$1+-3i$	$1+-2i$
Z	10	0	-10	**-20**	-30	-40	-50	41	31	21
Cz	$1+-1i$	$1+0i$	$1+1i$	$1+2i$	$1+3i$	$1+4i$	$1+5i$	$2+-4i$	$2+-3i$	$2+-2i$
Z	11	1	-9	-19	-29	-39	-49	**42**	32	22
Cz	$2+-1i$	$2+0i$	$2+1i$	$2+2i$	$2+3i$	$2+4i$	$2+5i$	$3+-4i$	$3+-3i$	$3+-2i$
Z	12	2	-8	-18	-28	-38	-48	43	33	23
Cz	$3+-1i$	$3+0i$	$3+1i$	$3+2i$	$3+3i$	$3+4i$	$3+5i$	$4+-4i$	$4+-3i$	$4+-2i$
Z	13	3	-7	-17	-27	-37	-47	44	34	24
Cz	$4+-1i$	$4+0i$	$4+1i$	$4+2i$	$4+3i$	$4+4i$	$4+5i$	$5+-4i$	$5+-3i$	$5+-2i$
Z	14	4	-6	-16	-26	-36	-46	45	35	25
Cz	$5+-1i$	$5+0i$								
Z	15	5								

где Z — множество целых чисел, а число $N > 0$ назовем *порядком комплексных дробей Фарея*.

Справедлива следующая теорема.

Теорема. Пусть N — порядок комплексных дробей Фарея, удовлетворяющий неравенству

$$4N^2 < \left\lceil \frac{p^2+q^2-1}{|p|+|q|} \right\rceil,$$

и пусть $x = z_1/z_2$ — комплексная дробь Фарея порядка N сравнима с некоторым целым комплексным числом k по модулю $M = p + iq$, тогда x — единственная дробь Фарея порядка N, сравнимая с k по модулю M.

Доказательство. Предположим существование двух дробей x, y, сравнимых по модулю M с k, где

$$x = \frac{a+ib}{c+id}, \quad y = \frac{a'+ib'}{c'+id'}.$$

Тогда

$$\left|\frac{a+ib}{c+id}\right|_M \equiv \left|\frac{a'+ib'}{c'+id'}\right|_M,$$

$$|(a+ib)(c'+id')|_M \equiv |(c+id)(a'+ib')|_M,$$

или

$$|(ac'-bd')+(bc'+ad')i|_M \equiv$$
$$\equiv |(ca'-b'd)+(da'+b'c)i|_M,$$

или

$$|((ac'-bd')+(ca'-b'd))+((bc'+ad')-$$
$$-(da'+b'c))i|_M \equiv 0.$$

Обозначим

$$a_1 = (ac'-bd')-(ca'-b'd),$$
$$b_1 = (bc'+ad')-(da'+b'c). \qquad (10)$$

Подставляя a_1, b_1 в формулу (4), получим

$$a_1p + b_1q \equiv 0 \pmod{p^2+q^2},$$
$$b_1p - a_1q \equiv 0 \pmod{p^2+q^2}.$$

Пусть выполняется неравенство

$$4N^2 < \left\lceil \frac{p^2+q^2-1}{|p|+|q|} \right\rceil. \qquad (11)$$

Рассмотрим первое сравнение (10). Тогда

$$|(a\cdot c' - b\cdot d')\cdot p + (b\cdot c' + a\cdot d')\cdot q -$$
$$-(c\cdot a' - b'\cdot d)\cdot p - (d\cdot a' + b'\cdot c)\cdot q| =$$
$$= |a\cdot c'\cdot p - b\cdot d'\cdot p - c\cdot a'\cdot p + b'\cdot d\cdot p +$$
$$+ b\cdot c'\cdot q + a\cdot d'\cdot q - d\cdot a'\cdot q -$$
$$- b'\cdot c\cdot q| \leqslant |a|\cdot|c'|\cdot|p| + |b|\cdot|d'|\cdot|p| + |c|\cdot|a'|\cdot|p| +$$
$$+ |b'|\cdot|d|\cdot|p| + |b|\cdot|c'|\cdot|q| + |a|\cdot|d'|\cdot|q| + |d|\cdot|a'|\cdot|q| +$$
$$+ |b'|\cdot|c|\cdot|q| \leqslant N^2\cdot|p| + N^2\cdot|p| + N^2\cdot|p| +$$
$$+ N^2\cdot|p| + N^2\cdot|q| + N^2\cdot|q| + N^2\cdot|q| +$$
$$+ N^2\cdot|q| = 4\cdot N^2\cdot|p| + 4\cdot N^2\cdot|q| \leqslant p^2+q^2-1,$$

что следует из (11).

Рассмотрим второе сравнение (10). Тогда

$$|(b\cdot c' + a\cdot d')\cdot p - (a\cdot c' - b\cdot d')\cdot q -$$
$$-(d\cdot a' + b'\cdot c)\cdot p + (c\cdot a' + b'\cdot d)\cdot q| =$$
$$= |b\cdot c'\cdot p + a\cdot d'\cdot p - a\cdot c'\cdot q + b\cdot d'\cdot q -$$
$$- d\cdot a'\cdot p - b'\cdot c\cdot p + c\cdot a'\cdot q -$$
$$- b'\cdot d\cdot q| \leqslant |b|\cdot|c'|\cdot|p| + |a|\cdot|d'|\cdot|p| + |a|\cdot|c'|\cdot|q| +$$
$$+ |b|\cdot|d'|\cdot|q| + |d|\cdot|a'|\cdot|p| + |b'|\cdot|c|\cdot|p| + |c|\cdot|a'|\cdot|q| +$$
$$+ |b'|\cdot|d|\cdot|q| \leqslant N^2\cdot|p| + N^2\cdot|p| + N^2\cdot|p| +$$
$$+ N^2\cdot|p| + N^2\cdot|q| + N^2\cdot|q| + N^2\cdot|q| +$$
$$+ N^2\cdot|q| = 4\cdot N^2\cdot|p| + 4\cdot N^2\cdot|q| \leqslant p^2+q^2-1,$$

что следует из (11).

Таблица 4

Co	$-2 + 0i/2 + 1i$	$-2 + 0i/1 + -2i$	$-2 + 0i/-1 + 2i$	$-2 + 0i/-2 + -1i$	$-2 + -1i/2 + 2i$
Z	312	176	225	89	295
Co	$-2 + -1i/2 + -2i$	$-2 + -1i/-2 + 2i$	$-2 + -1i/-2 + -2i$	$-2 + 2i/-1 + -2i$	$-2 + 2i/2 + -1i$
Z	115	286	106	184	71
Co	$-2 + 0i/1 + 0i$	$-2 + 2i/-2 + 1i$	$-2 + 2i/1 + 2i$	$-1 + -1i/2 + -1i$	$-1 + -1i/1 + 2i$
Z	399	330	217	92	236
Co	$-1 + -1i/-1 + -2i$	$-1 + -1i/-2 + 1i$	$-1 + -2i/-1 + 2i$	$-1 + -2i/2 + 1i$	$-1 + -2i/-2 + -1i$
Z	165	309	224	332	69
Co	$-1 + -2i/1 + -2i$	$-1 + 1i/1 + 0i$	$-1 + -1i/1 + 0i$	$-1 + 0i/1 + 0i$	$0 + 0i/1 + 0i$
Z	177	380	19	400	0
Co	$0 + 1i/2 + -2i$	$0 + 1i/1 + -2i$	$0 + 1i/-1 + -2i$	$0 + 1i/-2 + -2i$	$0 + 1i/2 + -1i$
Z	**95**	156	164	105	72
Co	$0 + 1i/-2 + -1i$	$0 + 1i/2 + 1i$	$0 + 1i/-2 + 1i$	$0 + 1i/2 + 2i$	$0 + 1i/1 + 2i$
Z	88	313	329	296	237
Co	$0 + 1i/-1 + 2i$	$0 + 1i/-2 + 2i$	$0 + 1i/0 + -2i$	$0 + 1i/-2 + 0i$	$0 + 1i/2 + 0i$
Z	245	306	200	10	391
Co	$0 + 1i/0 + 2i$	$0 + -1i/1 + 0i$	$0 + -2i/1 + 0i$	$0 + 2i/1 + 0i$	$0 + 1i/1 + 0i$
Z	201	20	40	**361**	**381**
Co	$1 + -1i/-1 + 2i$	$1 + -1i/2 + 1i$	$1 + -1i/-2 + -1i$	$1 + -1i/1 + -2i$	$1 + 0i/-1 + -1i$
Z	244	333	68	157	190
Co	$1 + 0i/-1 + 1i$	$1 + 1i/1 + 0i$	$1 + -1i/1 + 0i$	$1 + 0i/1 + -1i$	$1 + 0i/1 + 1i$
Z	**210**	382	21	191	**211**
Co	$1 + -2i/1 + 2i$	$1 + -2i/-2 + 1i$	$1 + -2i/2 + -1i$	$1 + -2i/-1 + -2i$	$1 + 0i/1 + 0i$
Z	256	**308**	93	145	1
Co	$2 + -1i/-2 + -2i$	$2 + -1i/-2 + 2i$	$2 + -1i/-1 + -1i$	$2 + -1i/-1 + 1i$	$2 + -1i/1 + -1i$
Z	85	**305**	170	209	192
Co	$2 + -1i/1 + 1i$	$2 + -1i/2 + -2i$	$2 + -1i/2 + 2i$	$2 + 0i/-2 + 1i$	$2 + 0i/-1 + -2i$
Z	**231**	96	316	**328**	144
Co	$2 + 0i/1 + 2i$	$2 + 0i/2 + -1i$	$2 + 1i/-1 + -1i$	$2 + 1i/-1 + 1i$	$2 + 1i/1 + -1i$
Z	257	73	189	230	171
Co	$2 + 1i/1 + 1i$	$2 + 2i/-2 + -1i$	$2 + 2i/-1 + 2i$	$2 + 2i/1 + -2i$	$2 + 0i/1 + 0i$
Z	212	87	265	136	2
Co	$2 + 2i/2 + 1i$				
Z	314				

Таким образом,

$$a_1 p + b_1 q \equiv 0 (\mathrm{mod}\ p^2 + q^2);$$
$$b_1 p - a_1 q \equiv 0 (\mathrm{mod}\ p^2 + q^2)$$

и

$$a_1 p + b_1 q < p^2 + q^2;$$
$$b_1 p - a_1 q < p^2 + q^2.$$

Но известно, что $|x|_{p^2+q^2} = x$, если $x < p^2 + q^2$. Отсюда получим, что

$$\begin{cases} a_1 p + b_1 q = 0, \\ b_1 p - a_1 q = 0. \end{cases}$$

Решая эту систему относительно a_1, b_1, получаем, что

$$a_1 = 0,\ b_1 = 0, \text{ т. е.}$$

$$(ac' - bd') + (bc' + ad')i = \\ = (ca' - b'd) + (da' + b'c)i$$

или

$$(a + ib)(c' + id') = (c + id)(a' + ib'),$$

т. е. $x = y$, что и требовалось доказать.

Алгоритмы преобразования вида {Комплексная дробь Фарея → в МС и МС → Комплексная дробь Фарея} принципиально ничем не отличаются от рассмотренных в работе [3]. На сайте [6] также размещены исходные тексты программ на Delphi, в которых эти алгоритмы реализованы.

В табл. 4 приведены комплексные дроби Фарея второго порядка и соответствующие им представления в модулярной системе счисления по модулю $M = 20 + i$ (соответствующий ему вещественный модуль равен $20^2 + 1 = 401$). В табл. 4 символом C_Q обозначены строки, содержащие комплексные дроби Фарея, а символом Z — соответствующие целые числа по модулю $p^2 + q^2 = 101$.

Найдем значение выражения D_1:

$$D_1 = \frac{2 - i}{1 + i} + i = \frac{2 - i + i - 1}{1 + i} = \frac{1}{1 + i}.$$

Найдем значение этого выражения, пользуясь табл. 4, получаем:

$$D_1 = (231 + 381) \mathrm{mod}\ 401 = 211.$$

Из табл. 4 видно, что **211** соответствует дробь $\frac{1}{1 + i}$.

Найдем значение выражения D_2:

$$D_2 = \frac{-2i}{1 + 2i} + i = \frac{-2i + i - 2}{1 + i} = \frac{-2 - i}{1 + i}.$$

В табл. 4 нет $\frac{-2i}{1 + 2i}$, так как эта дробь сократимая, а комплексные дроби Фарея — это несокра-

тимые дроби. Но $\frac{-2i}{1+2i} = \frac{2}{-2+i}$, поскольку $(-2i)(-2 + i) = 2(1 + 2i)$.

Пользуясь табл. 4, находим:

$$D_2 = (328 + 381)\text{mod } 401 = 308.$$

Из табл. 4 видно, что **308** соответствует дроби $\frac{1-2i}{-2+i} = \frac{-2-i}{1+i}$.

Найдем значение выражения

$$D_3 = \frac{2-i}{-2+2i} - \frac{1}{-1+i} = \frac{i+1}{(-2+2i)(-1+i)} = \frac{i+1}{-4i}.$$

Пользуясь табл. 4, находим:

$$D_3 = (305 - 210)\text{mod } 401 = 95,$$

что соответствует дроби $0 + 1i/2 + -2i\frac{i}{2-2i} \neq \frac{i+1}{-4i}$.

Ответ неверный, возникла ошибка псевдопереполнения, так как истинный искомый результат не есть комплексная дробь Фарея порядка 2.

Заключение

Приведенный в настоящей работе пример плохо обусловленной задачи второй категории — обращения матрицы с комплексными коэффициентами — показывает, что расчеты по формулам (3), выполняемые в арифметике с плавающей точкой, могут приводить к катастрофической потере точности результатов. В связи с этим в настоящей работе обобщена модель Грегори—Кришнамурти для реализации вычислений с исключением ошибок округления над полем комплексных чисел (с целой мнимой и вещественной частями), получены оценки модуля, необходимого для представления результатов. Полученные оценки модуля позволяют также определить целесообразность применения вычислений с исключением ошибок округления в конкретных ситуациях по критерию точность/быстродействие. Известно, что выбор очень большого модуля приводит к необходимости вычислений с целыми числами очень большой длины и от этого уменьшается быстродействие, а выбор малого модуля при выполнении арифметических операций очень быстро приведет к ошибке псевдопереполнения и к неверному результату. Поэтому оптимальным может быть вариант, когда часть вычислений проводится в арифметике с плавающей точкой, а другая часть, критичная к точности, — в "безошибочной" арифметике.

Список литературы

1. **Тихонов А. Н., Арсенин В. Я.** Методы решения некорректных задач. М.: Наука, 1979.
2. **Оцоков Ш. А.** Применение модулярной арифметики для решения непрерывной задачи наименьших квадратов // Инфокоммуникационные технологии в науке, производстве и образовании: Сб. матер. третьей международной научно-технической конференции. Ставрополь, 2008. С. 210—217.
3. **Грегори Р., Кришнамурти Е.** Безошибочные вычисления. Методы и приложения. М.: Мир, 1988. 207 с.
4. **Ясницкий Л. Н.** Введение в искусственный интеллект. М.: Изд. центр "Академия", 2005.
5. **Акушский Н. Я., Юдицкий Д. И.** Машинная арифметика в остаточных классах. М.: Сов. радио, 1968. 439 с.
6. **www.sitm.ru/modular**

УДК 004.9

В. И. Аникин, д-р техн. наук, проф., нач. отд.,
О. В. Аникина, ассистент,
Поволжский государственный университет сервиса, г. Тольятти, e-mail: anikin@tolgas.ru

Табличное моделирование клеточных автоматов в Microsoft Excel

Обсуждается новая технология табличного моделирования клеточных автоматов без программирования на языке VBA, базирующаяся на предложенной авторами оригинальной технике создания итерационных табличных моделей в Microsoft Excel. Результаты работы представляют интерес для специалистов по моделированию динамических систем и процессов, а также для преподавателей, аспирантов и студентов высших учебных заведений.

Ключевые слова: *клеточный автомат, Microsoft Excel, табличная модель, игра "Жизнь" Конуэя, "Муравей" Лэнгтона.*

Клеточные автоматы (КА), впервые введенные Дж. фон Нейманом и К. Цусе в середине XX века, нашли применение для распараллеливания вычислительных процессов, наглядной демонстрации явлений самоорганизации в нелинейных системах с локальными взаимодействиями, генерации красивых геометрических узоров, моделирования физических и химических явлений и процессов [1]. В последние годы появилось множество работ по практическому использованию нечетких клеточных автоматов [2] и решеток связанных отображений [3] для моделирования про-

странственно-временной динамики непрерывных систем и процессов различной природы (социальных, экономических, биологических, экологических и др.), как детерминированных, так и стохастических [4, 5].

В той или иной мере общим недостатком существующих программ моделирования КА являются закрытость программного кода и сложность расширения их функциональных возможностей. В этом смысле табличное моделирование КА, базирующееся на принципе "программирование без программирования", несомненно, является привлекательным и перспективным, требует от разработчика меньшей квалификации, предоставляет ему большую свободу действий, быстрее приводит к конечному результату.

Главные достоинства предлагаемой нами техники табличного моделирования клеточных автоматов заключаются в следующем:

- обеспечение свободы творчества и экспериментирования с клеточными автоматами для квалифицированных программистов и пользователей ПК, не обладающих хорошими программистскими навыками;
- полная открытость для анализа и модификации структуры, данных и алгоритма работы клеточного автомата;
- погруженность создаваемого КА в мощную вычислительную среду табличного процессора, обеспечивающую всесторонний анализ, эффективную отладку и удобное представление данных в виде таблиц, диаграмм, графиков;
- использование инструментальных средств и функциональных возможностей электронных таблиц для тестирования новых гипотез и идей, в том числе касающихся клеточных автоматов;
- исключительная полезность и наглядность табличных моделей КА для учебных целей.

Научная новизна данной работы заключается в том, что в ней предложена и реализована оригинальная техника создания итерационных табличных моделей, которые до сих пор никогда и никем не создавались. На примере клеточных автоматов убедительно доказывается, что электронные таблицы (ЭТ) могут применяться в качестве эффективной среды математического моделирования динамических систем, позволяющей в визуальном режиме, без программирования, имитировать структуру и алгоритмы работы последних.

Для демонстрации возможности табличной реализации КА рассмотрим два знаменитых клеточных автомата: компьютерная игра *Жизнь* Конуэя и *Муравей* Лэнгтона [http://ru.wikipedia.org]. Однако, прежде чем переходить к изложению основного материала статьи, укажем на ряд особенностей пересчета ячеек ЭТ в Excel, имеющих принципиальное значение для разработки итерационных табличных моделей [6].

1. Под итерацией в Excel понимается один цикл пересчета ячеек электронной таблицы. Чтобы разрешить множественные итерации, нужно на вкладке *Вычисления* окна параметров Excel выбрать флажок *Итерации*. Важно понимать, что использование неявных циклических ссылок, без которых создание полноценной табличной модели двумерных КА невозможно, допускается только при разрешенных итерациях.

2. Порядок автоматического пересчета ячеек, объединенных циклической ссылкой (и зависимых от них ячеек), отличается от порядка пересчета ячеек с обычными ссылками: ячейки с циклическими ссылками пересчитываются в последовательности слева направо и вниз по строкам ЭТ, тогда как ячейки с обычными ссылками пересчитываются, следуя связям между ними, как описано в справочной системе Excel.

3. Циклическую ссылку Excel обнаруживает динамически, непосредственно в ходе итераций. Например, вычисляя функцию ЕСЛИ(*условие; циклическая_ссылка; обычная_ссылка*), Excel фиксирует наличие циклической ссылки только при выполнении условия *условие = ИСТИНА*.

В этой связи табличную модель удобно дополнить двумя вспомогательными именованными ячейками: *N* (число итераций) и *Start* (флаг ручного включения итераций), а также управляющими кнопками *Пуск* и *НУ* (*Начальная установка*), ассоциировав с последними простые универсальные обработчики события *OnClick*:

```
'===== Кнопка Пуск =====
Private Sub StartBtn_Click()
  With Application
    '-- Пуск итераций --
    .Calculation =
    xlCalculationManual
    .Iteration = True
    Range("Start").Value = 1
    .MaxIterations =
    Range("N").Value
  ActiveSheet.Calculate
  End With
End Sub
```

```
'===== Кнопка НУ =====
Private Sub InitBtn_Click()
  With Application
    '-- Стоп итераций --
    .Calculation =
    xlCalculationManual
    .Iteration = True
    Range("Start").Value = 0
    .MaxIterations = 1

  ActiveSheet.Calculate
  End With
End Sub
```

Совместное использование в итерационной табличной модели флага ручного включения итераций *Start* и функции ЕСЛИ вида ЕСЛИ(*условие; циклическая_ссылка; обычная_ссылка*) позволяет пользователю по своему усмотрению (или программно) замыкать/размыкать контуры циклических связей непосредственно в ходе итераций.

Это простое, но эффективное нововведение дает существенные преимущества:

- возможность ручного управления итерациями непосредственно через пользовательский интерфейс модели;

- итерационная табличная модель приобретает удобный пользовательский интерфейс с кнопочным управлением;
- непосредственно перед итерациями и сразу же после их завершения макросы *StartBtn_Click()* и *InitBtn_Click()* могут выполнить некоторые дополнительные процедурные действия, например, на время выполнения итераций снять защиту листа, придать курсору мыши вид *Песочные часы*, сделать вспомогательную предили постобработку данных и т. д.

4. Для визуализации и анимации работы итерационных табличных моделей удобно пользоваться условным форматированием, которое, аналогично обнаружению циклической ссылки, в Excel также выполняется динамически.

Табличная модель клеточного автомата *Жизнь* Конуэя

Компьютерная игра *Жизнь* Конуэя разыгрывается на двумерной решетке клеток, каждая из которых может находиться в двух состояниях: 1 (живая) или 0 (мертвая). Состояния всех клеток меняются синхронно по шагам. На каждом следующем шаге новое состояние клетки определяется текущим состоянием ее окружения по правилам: 1) клетка, имеющая ровно три живых соседа из восьми, оживает (рождение); 2) клетка, имеющая два или три живых соседа, не изменяет своего состояния (выживание); 3) во всех остальных случаях клетка умирает (гибель от одиночества или перенаселения).

Далее, следуя великолепной программе М. Войтовича *Mcell* для исследования дискретных клеточных автоматов [http://www.mirekw.com], вместо модели игры *Жизнь* Конуэя создадим более общую табличную модель семейства клеточных автоматов *Жизнь*.

Табличная модель семейства КА *Жизнь* состоит из закрепленной области пользовательского интерфейса в интервале ячеек A1:BA6 (рис. 1) и пяти двумерных массивов клеток размером 50 × 50 ячеек для хранения информации о состоянии клеточного автомата (для краткости будем называть эти массивы полями клеток):

1) C8:AZ57 — входное поле *Начальное состояние КА* предназначено для задания детерминированного начального состояния клеточного автомата;

2) C59:AZ108 — поле *Инициализация КА*. При сброшенном флаге Хаос содержимое этого поля является копией поля C8:AZ57, при установленном флаге Хаос его ячейки заполняются нулями и единицами случайным образом;

3) C110:AZ159 — поле *Текущее состояние КА*, в каждой итерации определяет текущее состояние клеток автомата: 1 — живая клетка, 0 — мертвая клетка;

4) C161:AZ210 — вспомогательное поле *Состояние окружения*. В клетках этого поля рассчитывается число живых клеток, окружающих соответствующие сопряженные клетки поля C110:AZ159. Если установлен флаг Центр, то помимо состояния восьми ближайших соседей (окружение Мура [1]) учитывается также состояние центральной клетки;

5) C212:AZ261 — поле Копия состояния КА, является копией поля C110:AZ159.

Пользовательский интерфейс модели (см. рис. 1) включает:

- таблицу переходов КА в интервале ячеек F2:O4;
- управляющие флаги Хаос и Центр и связанные с ними ячейки Q1, Q3;
- зависимые от Q1, Q3 ячейки Q2, Q4, в которых по формулам [Q2] : = ЕСЛИ(Q1 = ИСТИНА; 1; 0) и [Q4] : = ЕСЛИ(Q3 = ИСТИНА; 1; 0) логические значения *ИСТИНА* и *ЛОЖЬ* флагов Хаос и Центр преобразуются в числовые значения 1 и 0;
- управляющие кнопки Пуск и НУ, с которыми ассоциированы универсальные макросы *StartBtn_Click()* и *InitBtn_Click()*;
- входные ячейки Вероятность жизни W2 и Число итераций AX2;
- ячейка *Флаг включения итераций* AX3, устанавливается и сбрасывается программно с помощью макросов *StartBtn_Click()* и *InitBtn_Click()*;
- ячейка *Счетчик итераций* AX4, используется для динамического счета текущего числа выполненных итераций по формуле [AX4] : = ЕСЛИ(Start = 1; AX4+1; 0).

Значения ячеек полей C59:AZ108, C110:AZ159, C161:AZ210 и C212:AZ261 рассчитываются по формулам:

[D60] : = ЕСЛИ(Q2 = 1;
ЕСЛИ(СЛЧИС() < W2; 1; 0); D9)

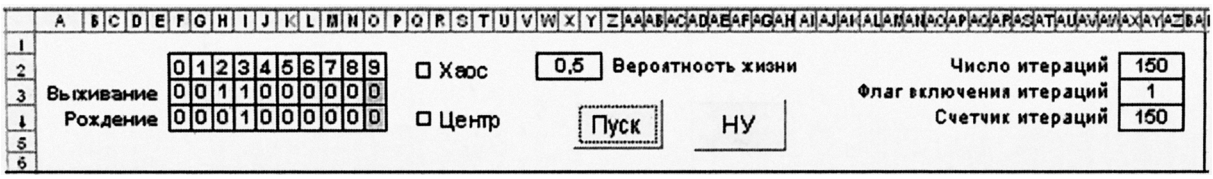


Рис. 1. Пользовательский интерфейс табличной модели семейства

[D111] : = ЕСЛИ(Start = 1; ЕСЛИ(И(СМЕЩ(E2; 2; D162+1) = 1; D213 = 0); 1; ЕСЛИ(СМЕЩ(E2; 1; D162+1) = 1; D213; 0)); D60)
[D162] : = СУММ(C110:E112) + (Q4-1)*D111
[D213] : = D111

С помощью маркера заполнения эти формулы копируются в остальные ячейки соответствующих полей КА. Отметим, что в модели используются симметричные граничные условия (противоположные границы полей клеток сшиваются, образуя двумерный тор), поэтому после операции копирования требуется скорректировать ссылки в граничных ячейках поля *Состояние окружения*.

Визуализация работы КА осуществляется посредством механизма условного форматирования Excel. Для этого нужно выделить ячейки поля *Текущее состояние КА* и с помощью команды меню *Формат/Условное форматирование* указать, чтобы клетки со значением 1 заливались черным цветом, а со значением 0 — белым.

Порядок работы с созданной табличной моделью семейства КА *Жизнь* следующий.

1. В интервал ячеек F3:O4 интерфейсной области модели вводится таблица переходов требуемого КА семейства *Жизнь* (на рис. 1 приведена таблица переходов для КА *Жизнь* Конуэя).

2. Задается детерминированное или случайное начальное состояние КА. Для задания детерминированного начального состояния КА необходимо сбросить флаг Хаос, в поле *Начальное состояние КА* ввести исходные значения клеток автомата и щелкнуть мышью на кнопке НУ. Для задания случайного начального состояния КА нужно выбрать флаг Хаос, в ячейке W2 задать вероятность жизни и щелкнуть мышью на кнопке НУ.

3. Модель КА запускается на выполнение щелчком мышью на кнопке Пуск. Предварительно в ячейке AX2 задается число итераций *N*, которое будет выполнено при прогоне модели.

Кнопки Пуск и НУ обеспечивают удобное и гибкое управление выполнением модельных экспериментов. Так, если после прогона модели вторично щелкнуть мышью на кнопке Пуск, то работа КА продолжится из достигнутого состояния, будет выполнен второй цикл итераций и т. д. В частности, задав в ячейке AX2 число итераций $N = 1$, модель можно выполнять в пошаговом режиме. И наоборот, щелкнув мышью на кнопке НУ, пользователь может принудительно остановить выполнение итераций и вернуть модель в исходное состояние.

На рис. 2 приведено конечное состояние табличной модели КА *Жизнь* Конуэя для начальной конфигурации *Планерное ружье*, полученное после выполнения 150 итераций.

Следует отметить, что программа М. Войтовича *Mcell* позволяет моделировать 13 различных семейств КА, и все эти семейства с небольшими видоизменениями мы успешно реализовали и в табличном варианте. При этом в сравнении с программной реализацией выявились два недостатка табличного моделирования КА в Excel: 1) меньший поперечный размер полей (разрешение) автомата; 2) более низкое быстродействие модели. К счастью, быстродействие табличной модели можно в несколько раз увеличить, скрыв визуализацию работы КА на время выполнения итераций.

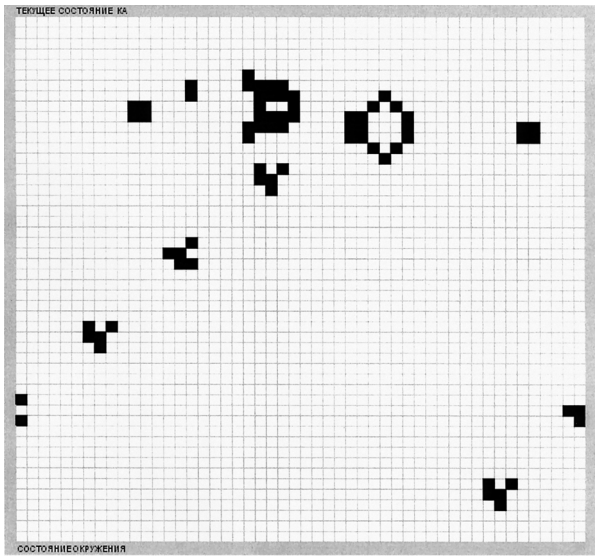


Рис. 2. Состояние КА *Жизнь* после 150 итераций для начальной конфигурации клеток *Планерное ружье*

Вместе с тем, в сравнении с программным моделированием табличное моделирование КА имеет очевидные преимущества, главными из которых являются быстрота и гибкость разработки новых клеточных автоматов. Например, как мы сейчас покажем, в Excel совсем нетрудно создать табличную модель клеточного автомата *Муравей* Лэнгтона — простейший пример нелинейного взаимодействия КА с другим автоматом (внешней средой), тогда как в программе *Mcell* подобную модель без дополнительного программирования реализовать невозможно.

Табличная модель клеточного автомата *Муравей* Лэнгтона

Средой обитания "муравья" является двумерное поле клеток черного и белого цветов. Муравей находится в одной из клеток этого поля. На каждом шаге он перемещается в одном из четырех направлений в клетку, соседнюю по стороне (окружение фон Неймана [1]), согласно следующим правилам:

- на черной клетке он поворачивается на 90° по часовой стрелке, затем изменяет цвет клетки среды обитания на противоположный и делает шаг вперед на соседнюю клетку;

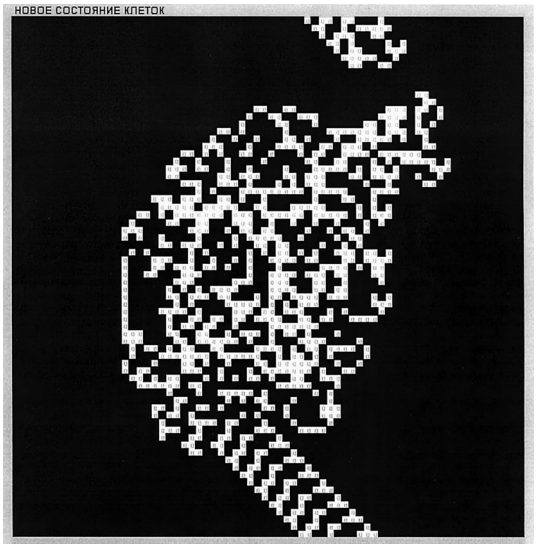


Рис. 3. Состояние среды обитания муравья после 11000 итераций

Таблица 1

Начальное состояние КА	Значение
Белая клетка без муравья	0
Черная клетка без муравья	1
Белая клетка с муравьем, направление движения →	2
Белая клетка с муравьем, направление движения ↓	3
Белая клетка с муравьем, направление движения ←	4
Белая клетка с муравьем, направление движения ↑	5
Черная клетка с муравьем, направление движения →	6
Черная клетка с муравьем, направление движения ↓	7
Черная клетка с муравьем, направление движения ←	8
Черная клетка с муравьем, направление движения ↑	9

Таблица 2

Состояние муравьев	Значение
В клетке нет муравья	0
В клетке есть муравей, направление движения →	1
В клетке есть муравей, направление движения ↓	2
В клетке есть муравей, направление движения ←	3
В клетке есть муравей, направление движения ↑	4

- на белой клетке он поворачивается на 90° против часовой стрелки, затем изменяет цвет клетки среды обитания на противоположный и делает шаг вперед на соседнюю клетку.

Эти простые правила вызывают довольно сложное поведение муравья в однородной среде: после длительного случайного блуждания он начинает двигаться по циклически повторяющейся траектории с периодом 104 шага, строя себе бесконечную прямоугольную "дорогу" (рис. 3).

Итерационная табличная модель КА *Муравей* Лэнгтона содержит пять полей размером 70 × 70 ячеек каждое:

1) ячейки поля G3:BX72 являются входными и предназначены для задания начального состояния муравья и среды обитания согласно табл. 1.

2) ячейки поля G74:BX143 предназначены для хранения нового состояния среды обитания муравья после очередного шага итераций: 1 — черная клетка, 0 — белая клетка;

3) ячейки поля G145:BX214 предназначены для хранения нового состояния муравья после очередного шага итераций согласно табл. 2.

4) ячейки полей G216:BX285 и G287:BX356 определяют состояния среды обитания и муравья перед очередным шагом итераций.

Значения ячеек этих полей рассчитываются по следующим формулам:
[G74] : = ЕСЛИ(Start = 1; ЕСЛИ(G287>= 1; ЕСЛИ(G216 = 0; 1; 0); G216; ЕСЛИ(ИЛИ(G3 = 1; G3>= 6); 1; 0)) (копируется во все ячейки интервала G74:BX143);
[H146] : = ЕСЛИ(Start = 1; ЕСЛИ(ИЛИ(И(I288 = 2; I217 = 1); И(I288 = 4; I217 = 0)); 3; ЕСЛИ(ИЛИ(И(H289 = 3; H218 = 1); И(H289 = 1; H218 = 0)); 4; ЕСЛИ(ИЛИ(И(G288 = 4; G217 = 1); H(G288 = 2; G217 = 0)); 1; ЕСЛИ(ИЛИ(И(H287 = 1; H216 = 1); И(H287 = 3; H216 = 0)); 2; ЕСЛИ(H287 > 0; 0; H287))))); ЕСЛИ(ИЛИ(H4 = 2; H4 = 6); 1; ЕСЛИ(ИЛИ(H4 = 3; H4 = 7); 2; ЕСЛИ(ИЛИ(H4 = 4; H4 = 8); 3; ЕСЛИ(ИЛИ(H4 = 5; H4 = 9); 4; 0))))) (копируется во все ячейки интервала G145:BX214, в граничных ячейках этого интервала, как и в КА *Жизнь*, ссылки в формулах должным образом корректируются);
[G216] : = G74 (копируется во все ячейки интервала G216:BX285);
[G287] : = G145 (копируется во все ячейки интервала G287:BX356).

Для визуализации временной динамики среды обитания и муравья в процессе выполнения модели также используется условное форматирование.

Пользовательский интерфейс табличной модели КА *Муравей* Лэнгтона проще интерфейса модели семейства КА *Жизнь* и включает входную ячейку *Число итераций*, именованные ячейки *Флаг включения итераций* и *Счетчик итераций*, управляющие кнопки Пуск и НУ с ассоциированными макросами *StartBtn_Click()* и *InitBtn_Click()*.

На рис. 3 показано состояние среды обитания КА *Муравей* Лэнгтона, полученное после выполнения 11000 итераций при следующем начальном состоянии КА: все клетки среды обитания черные, муравей находится в клетке AR42, направление движения ←.

В заключение статьи заметим, что последняя версия Excel 2007 содержит два важных улучшения, имеющих непосредственное отношение к табличному моделированию клеточных автоматов:

1) число столбцов на рабочем листе Excel увеличено с 256 до 16384, что практически снимает ограничение на предельный поперечный размер КА;

2) для заданного выделения теперь можно задавать произвольное число критериев условного форматирования (в предыдущих версиях Excel оно не превышало 3), что позволяет визуализировать динамику работы клеточных автоматов с большим числом состояний.

Список литературы

1. **Тоффоли Т., Марголус Н.** Машины клеточных автоматов. М.: Мир, 1991. 280 с.

2. **Cattaneo G., Flocchini P., Mauri G., Santoro N.** Cellular automata in fuzzy backgrounds. // Physica. 1997. D 105. P. 105—120.

3. **Kaneko K.** Theory and applications of coupled map lattices // New York: Willey, 1993. 195 p.

4. **Liu Y., Phinn S.** Modeling Urban Development with Cellular Automata Incorporating Fuzzy — set Approaches. // Computers, Environment and Urban Systems. 2003. N 27. P. 637—658.

5. **Li X., Yeh A. G.** Neural — network based cellular automata for simulating multiple land use changes using GIS. // International Journal of Geographical Information Science. 2002. N 16. P. 323—343.

6. **Аникин В. И., Аникина О. В.** Создание моделей коллективного поведения в среде Microsoft Excel. // Синергетика природных, технических и социально-экономических систем: Сборник статей Международной научно-технической конференции в 2 ч., ч. I. — Тольятти: Изд. ТГУС, 2007. С. 119—124.

ОПТИМИЗАЦИЯ

УДК 519.6

А. П. Карпенко, д-р физ.-мат. наук, проф.,
А. Н. Уторов, магистр,
В. Г. Федорук, канд. техн. наук, доц.,
МГТУ им. Н. Э. Баумана,
e-mail: Karpenko@nog.ru

MPI-балансер загрузки многопроцессорной вычислительной системы для решения задачи многокритериальной оптимизации

Работа посвящена многофункциональному MPI-балансеру загрузки многопроцессорной вычислительной системы для решения задачи многокритериальной оптимизации. Балансер реализует один статический и три динамических метода балансировки. Рассматривается структура балансера, приводятся результаты исследования его эффективности, а также пример использования.

Ключевые слова: *статическая балансировка загрузки, динамическая балансировка загрузки, вычислительный кластер, MPI.*

Введение

При оптимизации сложных технических систем возникают задачи непрерывной многокритериальной оптимизации (МКО-задачи), имеющие следующие особенности:

- высокая размерность вектора альтернатив;
- сложная структура множества допустимых альтернатив, обусловленная большим числом и нелинейностью ограничивающих функций, формирующих это множество;
- высокая размерность критериальной вектор-функции;
- высокая сложность математических моделей оптимизируемых технических систем, приводящая к высокой вычислительной сложности частных критериев;
- сложная топология частных критериев оптимальности (овражность, многоэкстремальность и пр.) [1].

Указаные особенности современных МКО-задач требуют использования для их решения многопроцессорных вычислительных систем, например вычислительных кластеров.

Методы решения МКО-задач чрезвычайно разнообразны (что является, в конечном счете, следствием плохой формализуемости таких задач). По одной из распространенных классификаций выделяются следующие классы этих методов:

- методы зондирования;
- априорные методы;
- апостериорные методы;
- адаптивные методы.

Методы всех классов, кроме первого, сводят, в конечном счете, решение многокритериальной задачи к решению однокритериальной задачи глобальной условной оптимизации (ОКО-задачи). Наиболее распространенные подходы к решению ОКО-задачи основаны на сочетании локальных детерминированных поисковых методов и методов случайного поиска [2].

Рассматриваемый балансер загрузки ориентирован па параллельное решение МКО-задачи од-

ним из методов локальной однокритериальной оптимизации с использованием в качестве начальных точек узлов случайной сетки, покрывающей множество допустимых альтернатив. Однако балансер может быть использован для решения и любых других задач, информационная модель которых может быть представлена в виде двухуровневого дерева, листья которого имеют случайные вычислительные сложности с неизвестными статистическими характеристиками. В виде такой модели могут быть представлены, например, модели задачи вычисления кубатур, задачи построения некоторыми методами множества Парето в МКО-задаче и пр.

Балансер реализует следующие методы балансировки загрузки:

- статическая балансировка;
- динамическая централизованная балансировка с использованием равномерной декомпозиции узлов — равномерная балансировка;
- динамическая централизованная балансировка с использованием экспоненциальной декомпозиции узлов — экспоненциальная балансировка;
- динамическая децентрализованная диффузная балансировка с перераспределением загрузки по инициативе получателя — диффузная балансировка [3, 4].

В последнем случае рассматривается перераспределение загрузки только по инициативе получателя, поскольку для рассматриваемого класса задач такой алгоритм более эффективен, чем алгоритм перераспределения загрузки по инициативе отправителя. Детальное описание балансера приведено в работе [5].

1. Постановка задачи и ее информационная модель

Пусть $X \in R^n$ — n-мерный вектор варьируемых параметров, где R^n — n-мерное арифметическое пространство. Вектор X определен в непустом "технологическом" параллелепипеде

$$\Pi = \{X | x_i^- \leqslant x_i \leqslant x_i^+,\ i \in [1{:}n]\} = \\ = \{X | g_j(X) \geqslant 0,\ j \in [1{:}2n)\}.$$

На вектор X может быть дополнительно наложено некоторое число ограничений, формирующих множество

$$D = \{X | g_k(X) \geqslant 0,\ k = 2n + 1, 2n + 2, ...]\}.$$

Множеством допустимых значений вектора X является замкнутое множество

$$D_X = \Pi \cap D = \{X | G(X) \geqslant 0\},$$

где $G(X) = \{g_1(X),\ g_2(X),\ ...,\ g_{2n},\ g_{2n+1}(X),\ g_{2n+2}(X), ...\}$ — ограничивающая вектор-функция.

Положим, что $\Phi(X) = (\phi_1(X), \phi_2(X), ..., \phi_m(X))$ — векторный критерий оптимальности, определенный на параллелепипеде Π, со значениями в пространстве R^m. Лицо, принимающее решения, стремится минимизировать каждый из частных критериев оптимальности $\phi_1(X)$, $\phi_2(X)$, ..., $\phi_m(X)$.

Во введенных обозначениях рассматриваемую МКО-задачу *условно* запишем в виде

$$\min_{X \in D_X} \Phi(X) = \Phi(X^*), \qquad (1)$$

где X^* — искомое решение задачи.

Положим, что тем или иным методом решение задачи (1) сведено к решению ОКО-задачи

$$\min_{X \in D_X} \varphi(X) = \varphi(X^*), \qquad (2)$$

где $\varphi(X)$ — некоторая скалярная функция, например, аддитивная скалярная свертка вида $\varphi(X) = \varphi(\Lambda, X) = \sum_{i=1}^{m} \lambda_i \phi_i(X)$. Здесь $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_m)$ — вектор весовых множителей.

Чаще всего задачу глобальной условной оптимизации (2) сводят с помощью метода штрафных функций к задаче глобальной безусловной оптимизации

$$\min_{X \in R^n} f(\alpha, X) = \min_{X \in R^n} (\varphi(X) + \chi(\alpha, X)) = f(\alpha, X^*), \qquad (3)$$

где $f(\alpha, X)$ — функция, которая возрастает вблизи границ области допустимых значений D_X и тем быстрее, чем больше значение параметра α штрафной функции $\chi(\alpha, X)$. В качестве приближенного решения задачи (2) принимается решение X^* вспомогательной задачи (3) при достаточно большом α.

Схему приближенного решения задачи (2) комбинацией локального детерминированного поискового метода и метода случайного поиска можно представить в следующем виде.

1. Покрываем параллелепипед Π случайной сеткой Δ с узлами $X_1, X_2, ..., X_\varsigma$.

2. Проверяем принадлежность узла X_j, $j = 1, 2, ..., \varsigma$ множеству D_X.

3. Если $X_j \in D_X$, то исходя из этой точки методом локального поиска решаем задачу (2) — находим точку X_j^*.

4. Среди всех найденных точек $X_{j_1}^*$, $X_{j_2}^*$, ... находим точку, в которой значение функции $\varphi(X)$ является минимальным, и принимаем эту точку в качестве приближенного решения задачи.

Некоторые методы многокритериальной оптимизации могут требовать решения на каждой ите-

рации не одной, а некоторой совокупности ОКО-задач. Например, прямые адаптивные методы решения таких задач [6] требуют на каждой итерации поиска глобального минимума функции $\varphi(\Lambda, X)$, полученной при различных значениях вектора весовых множителей Λ, т. е. решения совокупности ОКО-задач вида

$$\min_{X \in D_X} \varphi(\Lambda_i, X) = \varphi(\Lambda_i, X_i^*),\ i \in [1{:}Z]. \qquad (4)$$

Задачи (4) могут решаться разными методами локальной оптимизации и на разных сетках, покрывающих параллелепипед Π: при $\Lambda = \Lambda_i$ задача решается на сетке Δ_i, с ς_i узлами $X_{i,1}, X_{i,2}, ..., X_{i,\varsigma_i}$; $i \in [1{:}Z]$. Такая схема решения очевидным образом сводится к последовательному применению рассмотренной выше схемы. Однако при этом теряется значительная часть параллелизма. Поэтому для решения задач (4) целесообразно использовать схему решения, которой соответствует граф Γ_1 потока данных в виде дерева, представленного на рис. 1.

Объединим в графе Γ_1 вершины так, как показано на рис. 1, и введем обозначения $X_1 = X_{1,1}$, $X_2 = X_{1,2}$ и т. д. до $X_\varsigma = X_Z$, ς_z, где $\varsigma = \sum_{i=1}^{Z} \varsigma_i$. В результате получим граф потока данных Γ_2 в виде двухуровневого дерева, которое используется в работе в качестве информационной модели задачи (рис. 2).

С точки зрения организации параллельных вычислений существенной особенностью этого дерева является то, что вычислительные сложности каждой из вершин *2* заранее неизвестны и могут изменяться в очень широком диапазоне. Причин

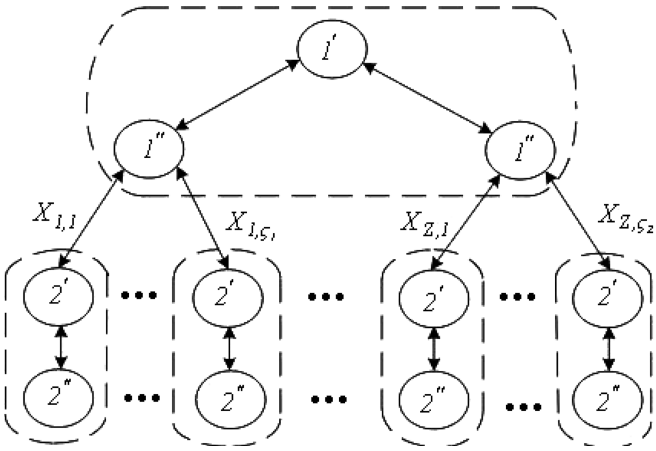


Рис. 1. Граф Γ_1 потока данных задачи (1), (2):
1′ — решение МКО-задачи; *1″* — глобальная оптимизация; *2′* — локальная оптимизация; *2″* — вычисление значений функций $G(X)$, $\Phi(X)$

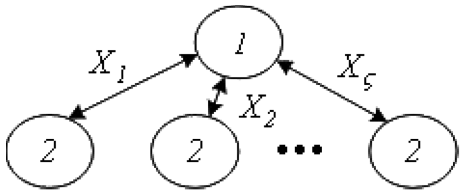


Рис. 2. Граф Γ_2 потока данных — информационная модель задачи

этому две. Во-первых, если $X_j \notin D_X$, $j \in [1{:}\varsigma]$, то приходится вычислять значения вектор-функции $G(X)$ и не вычислять значения критерия оптимальности $\Phi(X)$. Во-вторых, если $X_j \in D_X$, то число итераций, необходимых для отыскания локального минимума функции $\varphi(X)$, зависит от того, насколько "далеко" узел X_j находится от ближайшего локального минимума этой функции. Если для решения ОКО-задачи (2) используется метод штрафных функций, то вычислительная сложность решения существенно зависит также от того, насколько "далеко" ближайшая точка локального минимума функции $f(\alpha, X)$ находится от границы области допустимых значений D_X.

В качестве вычислительной системы рассматривается однородная система с распределенной памятью, состоящая из процессоров $P_1, P_2, ..., P_N$ и host-процессора.

2. Используемые методы балансировки

Статическая балансировка

Шаг 1. Host-процессор строит сетки Δ_i, $i \in [1{:}Z]$, и разбивает ς узлов $X_1, X_2, ..., X_\varsigma$ этих сеток на N множеств Ω_j, $j \in [1{:}N]$, не более чем по $s = [\varsigma/N]$ узлов в каждой. Здесь и далее $[\cdot]$ — символ ближайшего целого большего.

Шаг 2. Процессор P_k, $k \in [1{:}N]$, принимает от host-процессора координаты узлов множества Ω_j, исходя из каждого из этих узлов решает задачу локальной оптимизации и передает host-процессору результаты вычислений.

Шаг 3. Host-процессор на основе этих результатов находит приближенное решение задачи X^*, $\Phi(X^*)$.

Динамическая равномерная балансировка

Шаг 1. Host-процессор строит сетки Δ_i, $i \in [1{:}Z]$, и разбивает их ς узлов на K множеств Ω_j, $j \in [1{:}K]$, не более чем по $s = [\varsigma/K]$ узлов в каждой.

Шаг 2. Процессор P_k, $k \in [1{:}N]$, принимает от host-процессора координаты узлов первого из нераспределенных множеств Ω_j, обрабатывает их и передает host-процессору результаты вычислений.

Шаг 3. Если исчерпаны не все множества Ω_j, то host-процессор посылает, а процессор P_k прини-

мает координаты следующего нераспределенного множества узлов, которое обрабатывается процессором P_k аналогично шагу 2 и т. д.

Шаг 4. Host-процессор находит приближенное решение задачи.

Динамическая экспоненциальная балансировка

Шаг 1. Host-процессор строит сетки Δ_i, $i \in [1{:}Z]$, и полагает $k = 1$, $\xi_1 = \varsigma$.

Шаг 2. Если исчерпаны не все ς узлов, то host-процессор выделяет среди оставшихся ξ_k узлов множество Ω_k, содержащее $z_k = [\mu\xi_k]$ узлов, и разбивает это множество на N подмножеств $\varpi_{k,j}$, $j \in [1{:}N]$, не более чем по $s_k = [z_k/N]$ узлов в каждом; $0 < \mu < 1$.

Шаг 3. Пока множество узлов Ω_k не исчерпано, host-процессор передает, а процессор P_l, $l \in [1, N]$, принимает от него координаты узлов первого из необработанных подмножеств $\varpi_{k,j}$, обрабатывает их и передает host-процессору результаты вычислений.

Шаг 4. Если множество узлов Ω_k исчерпано, то host-процессор полагает $k = k + 1$, $\xi_k = \xi_k - z_k$ и переходит к шагу 2.

Шаг 5. Host-процессор находит приближенное решение задачи.

Диффузная балансировка

Шаг 1. Host-процессор строит сетки Δ_i, $i \in [1{:}Z]$, и разбивает их узлы на N множеств Ω_j, $j \in [1{:}N]$, не более чем по $s = [\varsigma/N]$ узлов в каждом.

Шаг 2. Процессор P_k, $k \in [1{:}N]$, принимает от host-процессора координаты узлов множества Ω_1.

Шаг 3. Процессор P_k выполняет следующие действия. Проводит обработку распределенных ему узлов и после завершения обработки передает host-процессору результаты обработки. Посылает запрос нескольким ближайшим процессорам P_{l_1}, P_{l_2}, ... Получает в ответ от каждого их этих процессоров число еще не обработанных ими узлов. Посылает запрос тому из указанных процессоров, который имеет наибольшее число необработанных узлов. Получает от этого процессора ν-ю часть его необработанных узлов и начинает их обработку ($0 < \nu < 1$).

Шаг 4. Если процессор P_l еще не закончил обработку распределенных ему узлов и получил первый запрос от процессора P_k, то он посылает ему в ответ число еще необработанных узлов.

Шаг 5. Если процессор P_l получил повторный запрос от процессора P_l, то он посылает в ответ этому процессору ν-ю часть своих еще не обработанных узлов (но не менее чем p узлов).

Шаг 6. Если host-процессор получил все необходимые результаты, то он посылает всем slave-процессорам сигнал предварительного завершения работы.

Шаг 7. Если процессор P_k получил от host-процессора сигнал предварительного завершения работы, то он высылает host-процессору подтверждение и перестает отправлять запросы другим процессорам (но не престает отвечать на них).

Шаг 8. Получив от всех slave-процессоров подтверждение о готовности к завершению, host-процессор посылает им сигнал окончательного завершения работы.

Шаг 9. Если процессор P_k получил от host-процессора сигнал окончательного завершения работы, то он завершается.

Шаг 10. Host-процессор находит приближенное решение задачи.

3. Состав и общая схема функционирования балансера

Балансер состоит из модулей статической, динамической равномерной, динамической экспоненциальной и динамической диффузной балансировки. Каждый из модулей поддерживает две группы функций: функции, отвечающие за работу host-процессора; функции, отвечающие за работу slave-процессора.

Как на host-процессоре, так и на каждом из slave-процессоров запускается по два процесса — пользовательский (вычислительный) процесс и управляющий процесс, реализующий функции балансера. Полагается, что host-пользовательский процесс реализует функции *1′*, *1″*, а slave-пользовательские процессы — функции *2′*, *2″* (см. рис. 1). Host- и slave-пользовательские процессы средствами балансера смогут взаимодействовать только со "своими" управляющими процессами (рис. 3). В первых трех модулях балансера возможны толь-

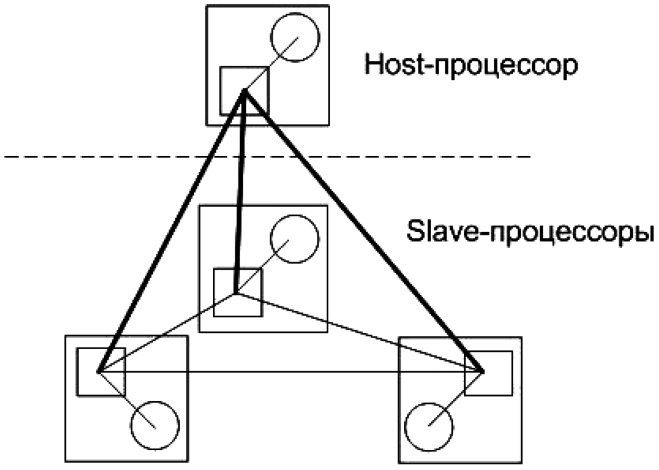


Рис. 3. Взаимодействие процессов в МВС, содержащей host-процессор и три подчиненных процессора:
окружность — вычислительный процесс; квадрат — управляющий процесс

ко двунаправленные обмены данными между host- и slave-управляющими процессами, а в четвертом модуле — также и обмены между slave-управляющими процессами.

Заметим, что для балансера host-процессор является таким же узлом кластера, как и остальные процессоры, — в качестве host-процессора балансер выбирает первый из списка процессоров, на которых запущена MPI-программа балансера.

Host-балансер получает от host-пользовательской программы следующие данные:

- используемый метод балансировки;
- общее число Z расчетных сеток Δ_i;
- значения величин n, m;
- общее число узлов ς;
- для каждого из указанных узлов номер i соответствующей расчетной сетки Δ_i, номер узла, n его координат.

Host-балансер в соответствии с заданным методом балансировки рассчитывает для каждого из slave-процессоров число узлов, которые должна обработать соответствующая slave-пользовательская программа. Затем host-балансер передает необходимые данные каждому из slave-балансеров. Slave-балансеры получают от host-балансера эти данные и передают их "своим" slave-пользовательским программам (за исключением используемого метода балансировки).

После завершения slave-пользовательскими программами обработки назначенных им узлов каждый из slave-балансеров получает от slave-пользовательской программы и передает host-баласеру две следующие группы данных:

- общее число успешно обработанных узлов (узлов, исходя из которых пользовательской программе удалось получить решение задачи). Для каждого из таких узлов — номер расчетной сетки; номер узла; n компонентов вектора X^*, соответствующего найденному решению; m компонентов этого решения; оценку вычислительной сложности полученного решения (которая может быть использована для оптимизации вычислительного процесса);
- общее число "дефектных" узлов (узлов, исходя из которых пользовательской программе по той или иной причине не удалось получить решение задачи). Для каждого из таких узлов — номер расчетной сетки, номер узла.

Host-балансер передает host-пользовательской программе указанные выше группы данных.

Входные данные балансера объединены во входные файлы: *файл параметров балансировки, входной файл*. Выходные данные объединены в выходные файлы: *файл числа результирующих точек; выходной файл; файл "дефектных" узлов; файл отчета; файл ошибок задания* [5].

Запуск балансера выполняется по следующей схеме:

- загрузка *файла конфигурации кластера*;
- загрузка *файла текущего использования кластера*;
- анализ *файла задания на корректность входных данных*; если данные не корректны, то пользователю выдается код ошибки и формируется файл ошибок задания;
- формирование команд запуска балансера;
- последовательный запуск сформированных команд на исполнение [5].

Разработка балансера выполнена с использованием компилятора *gcc* и библиотеки MPICH. Балансер работает под управлением операционной системы Linux и требует около 2800 Кбайт свободного пространства на жестком диске. Объем требуемой оперативной памяти определяется сложностью решаемой задачи.

4. Тестирование и оценка эффективности балансера

Тестирование балансера и оценка его эффективности выполнены на вычислительном кластере МГТУ им. Н. Э. Баумана, который состоит из 12 двухпроцессорных узлов на базе микропроцессора Pentium III 800 МГц, объединенных коммуникационной сетью в двумерный тор.

Было выполнено несколько серий экспериментов [5]. Во всех случаях использовались следующие соглашения:

- число расчетных сеток $Z = 1$;
- вычислительная сложность вектор-функции $G(X)$ равна нулю;
- вычислительная сложность C_f вектор-функции $\Phi(X)$ является случайной величиной, равномерно распределенной в интервале $[0, C_f^{\max}]$;
- мерой эффективности балансера является ускорение $S = T_1/T_N$, где T_1, T_N — время решения задачи на одном и N процессорах соответственно;
- моделирование величины C_f выполняется путем выполнения $C_f/2$ сложений и $C_f/2$ умножений вещественных чисел.

Тестирование показало работоспособность и высокую надежность балансера. Результаты тестирования балансера показывают, что при невысокой вычислительной сложности C_f ускорение существенно отличается от N. Как и следовало ожидать, с ростом вычислительной сложности вследствие относительного снижения накладных расходов эффективность балансировки повышается. Во всех экспериментах балансер обеспечивает существенно более высокую эффективность при использовании динамической равномерной и динамической экспоненциальной балансировки. Эффективность статической балансировки в ис-

следованном диапазоне вычислительных сложностей C_f оказывается выше эффективности диффузной балансировки, хотя логично было бы ожидать противоположного результата. Этот эффект объясняется следующими основными причинами:

- использование в качестве параметра балансировки ρ (см. п. 2) достаточно большого значения — 10. Если вычислительные сложности оставшихся необработанными узлов достаточно велики, то это обстоятельство может привести к значительному дисбалансу загрузки процессоров;
- выбор "соседних" процессоров без учета топологии коммуникационной сети. В результате этими процессорами могут оказаться не соседние процессоры, что приводит к росту коммуникационных расходов;
- ответы диффузного балансера на запросы, поступившие от соседних процессоров, с некоторой задержкой — через определенный период времени. При отсутствии узлов для расчета диффузный балансер с этим же периодом опрашивает соседние процессоры. Таким образом, если на одном из соседних процессоров имеются узлы, доступные для передачи данному процессору, то он начнет их обработку только через 2—3 указанных временных периода;
- большие коммуникационные расходы — диффузный балансер периодически опрашивает и отвечает на запросы других балансеров, что требует соответствующего процессорного времени.

5. Пример использования балансера

Рассмотрим задачу о вертикальном подъеме ракеты-зонда [7]. После нормирования движение ракеты описывается системой обыкновенных дифференциальных уравнений

$$\begin{cases} \dot{y}_1 = -u; \\ \dot{y}_2 = y_3; \\ \dot{y}_3 = -g + \dfrac{1}{y_1(t)}\left(Vu - C\mathbf{e}^{-\gamma y_2(t)} y_3^2(t)\right). \end{cases} \quad (4)$$

Здесь $y_1(t)$ — масса ракеты; $y_2(t)$ — высота подъема ракеты; $y_3(t)$ — вертикальная скорость ракеты; V — постоянная, характеризующая реактивную тягу двигателя ракеты; $t \in [0, T]$ — время; g, C, γ — постоянные, связанные с силой тяготения, аэродинамическим сопротивлением и убыванием плотности воздуха с высотой соответственно. Управление $u(t) \in d_U$ определяет режим расхода горючего, где множество допустимых управлений $d_u = \{u(t) | 0 \leqslant u(t) \leqslant u^+\}$; u^+ — известная константа. В качестве начальных значений переменных состояния рассматриваются величины $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 0$. Задача решается при следующих значениях параметров: $T = 100$; $u^+ = 0{,}04$; $g = 0{,}01$; $V = 2{,}0$; $C = 0{,}05$; $\gamma = 0{,}01$.

В постановке [7] задача состоит в определении управления $u(t)$, обеспечивающего минимум критерия оптимальности $(-y_2(T))$ при условии $y_1(T) = 0{,}2$, — задача подъема ракеты за фиксированное время на максимальную высоту при заданном запасе горючего. Мы рассматриваем двухкритериальную модификацию этой задачи: найти управление $u(t)$, которое минимизирует критерий оптимальности $(-y_2(T))$ и критерий оптимальности $(-y_1(T))$. Такая постановка формализует задачу подъема ракеты-зонда за фиксированное время на максимальную высоту при максимальном запасе горючего.

Для решения поставленной двухкритериальной задачи оптимального управления используется метод сведения этой задачи к задаче нелинейного программирования. Хорошо известно, что этот метод обладает серьезными недостатками [7], однако для наших целей это обстоятельство не существенно.

Покроем интервал $t \in [0, T]$ сеткой $t_0 = 0$, $t_1, t_2, \ldots, t_n$ и аппроксимируем на интервале $t \in [0, T]$ управление $u(t)$ кусочно-постоянной функцией со следующими значениями: $u(t) = u_1$, $t \in [0, t_1)$; $u(t) = u_2$, $t \in [t_1, t_2)$; ... $u(t) = u_n$, $t \in [t_{n-1}, t_n]$. Введем в рассмотрение n-мерный вектор $U = (u_1, u_2, \ldots, u_n)$. Критерии оптимальности при этом приобретут вид $\phi_1(u) = -y_2(T)$, $\phi_2(u) = -y_1(T)$, а множество допустимых управлений d_u трансформируется в множество $D_U = \{u_i | 0 \leqslant u_i \leqslant u^+,\ i \in [1{:}n]\}$, представляющее собой параллелепипед в пространстве R^n.

Таким образом, МКО-задача в данном случае *условно* записывается в виде

$$\min_{U \in D_U} \Phi(U) = \Phi(U^*), \quad (5)$$

где $\Phi(U) = (\phi_1(U), \phi_2(U))$ — векторный критерий оптимальности; U^* — искомое приближенное решение этой задачи.

С помощью аддитивной скалярной свертки $\varphi(U) = \lambda_1\phi_1(U) + \lambda_2\phi_2(U)$ сведем задачу (5) к ОКО-задаче

$$\min_{U \in D_U} \varphi(U) = \varphi(U^*). \quad (6)$$

В свою очередь, с помощью метода штрафных функций сведем задачу (6) к задаче глобальной безусловной оптимизации

$$\min_{U \in R^n} f(\alpha, U) = \min_{U \in R^n} (\varphi(U) + \chi(\alpha, U)) = f(\alpha, U^*), \quad (7)$$

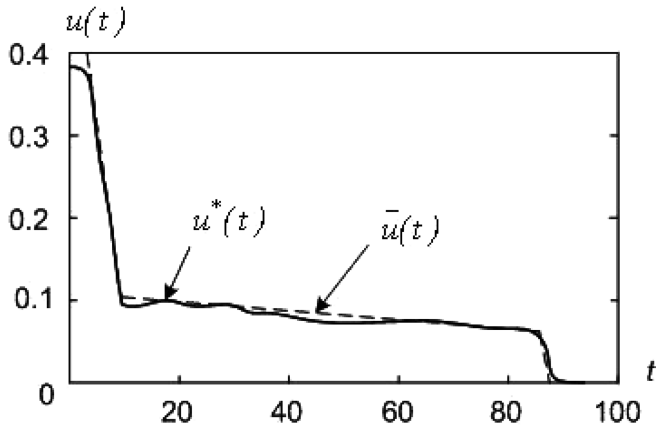


Рис. 4. Точное решение $\bar{u}(t)$ и найденное приближенное решение $u^*(t)$

где в качестве штрафной функции используем функцию

$$\chi(\alpha, U) = \sum_{i=1}^{n} \alpha_i (g_i^+(U))^2.$$

Здесь

$$g_i^+(U) = \begin{cases} 0, & u_i \in [0, u^+]; \\ u_i, & u_i \notin [0, u^+]. \end{cases}$$

Для решения задачи (7) используем комбинацию детерминированного метода локального поиска Нелдера—Мида и метода случайного поиска. Размерность вектора U положим равной 100 ($n = 100$); весовые множители в аддитивной свертке $\lambda_1 = 0{,}8$, $\lambda_2 = 0{,}2$. В случайной сетке, покрывающей параллелепипед D_U, примем $Z = 1000$ узлов. Параметры метода Нелдера—Мида положим равными $\alpha = 3$, $\beta = 0{,}5$, $\gamma = 0{,}5$. Начальную длину ребра симплекса примем равной $L = 0{,}01$ (четверть от максимально допустимого значения управления); требуемую точность решения $p = 0{,}01$.

Решение задачи выполнено на упомянутом в п. 4 вычислительном кластере МГТУ им. Н. Э. Баумана. Использовалось восемь процессоров. Результат решения задачи представлен на рис. 4. Рисунок показывает, что найденное решение $u^*(t)$ близко к точному решению $\bar{u}(t)$, полученному в книге [7] для однокритериальной задачи (что объясняется малым весом критерия оптимальности $\phi_2(u)$). Время решения задачи на одном процессоре составило 16 ч 35 мин, на 8 процессорах — 3 ч 20 мин. Таким образом, на восьми процессорах получено ускорение, равное примерно 5.

Заключение

Рассмотренный в работе многофункциональный MPI-балансер загрузки реализует один статический и три динамических метода балансировки. Балансер ориентирован на класс задач, информационная модель которых может быть представлена в виде двухуровневого дерева, листья которого имеют случайные вычислительные сложности с неизвестными статистическими характеристиками. Примерами таких задач являются задачи многокритериальной оптимизации и однокритериальной глобальной оптимизации, решаемые комбинацией одного из методов локального поиска и метода случайного поиска.

Тестирование балансера и оценка его эффективности выполнены на 12-процессорном вычислительном кластере МГТУ им. Н. Э. Баумана. Тестирование показало работоспособность и высокую надежность балансера.

В качестве примера рассмотрено использование балансера для решения двухкритериальной задачи оптимизации, которая формализует задачу подъема ракеты-зонда за фиксированное время на максимальную высоту при максимальном запасе горючего.

В продолжение работы планируется исследовать вклад факторов, перечисленных в п. 4, снизить эффективности диффузного балансера и на этой основе выполнить его модификацию.

Список литературы

1. **Карпенко А. П., Федорук В. Г.** Обзор программных систем многокритериальной оптимизации. Отечественные системы // Информационные технологии. 2008. № 1. С. 15—22.

2. **Лотов А. В.** Введение в экономико-математическое моделирование. М.: Наука, 1984. 392 с.

3. **Карпенко А. П., Федорук В. Г.** Балансировка загрузки многопроцессорной вычислительной системы при распараллеливании одного класса вычислительных задач // Тр. Всероссийской научн. конф. "Научный сервис в ИНТЕРНЕТ: многоядерный компьютерный мир". М.: Изд-во МГУ, 2007. С. 48—52.

4. **Карпенко А. П., Федорук В. Г., Федорук Е. В.** Балансировка загрузки многопроцессорной системы при распараллеливании одного класса вычислительных задач // Информационные технологии. 2008. № 3. С. 17—24.

5. **Карпенко А. П., Уторов А. Н., Федорук В. Г.** Многофункциональный балансер загрузки для решения одного класса вычислительных задач // Наука и образование: электронное научное издание. Инженерное образование. (№ Гос. регистрации 0420800025, ЭЛ № ФС 77-305 69), июнь, 2008. http://technomag.edu.ru.

6. **Карпенко А. П., Федорук В. Г.** Один класс прямых адаптивных методов многокритериальной оптимизации // Информационные технологии, 2009. № 5. С. 24—30.

7. **Федоренко Р. П.** Приближенное решение задач оптимального управления. М.: Наука, 1978. 488 с.

УДК 519.854.2, 519.8:658

Ю. А. Мезенцев, канд. экон. наук, доц.,
Новосибирский государственный
технический университет

Оптимизация расписаний параллельно-последовательных систем в календарном планировании

Предложены модели и алгоритм синтеза оптимальных расписаний параллельно-последовательных обслуживающих систем. Представленные модели синтеза расписаний принадлежат к классу линейных задач оптимизации с булевыми переменными либо редуцируются в него, ориентированы на практическое применение в календарном планировании и оперативном регулировании производственных процессов с дискретным характером.

Ключевые слова: *оптимизация, календарное планирование, теория расписаний, обслуживающие системы, параллельно-последовательные системы.*

Введение

Наиболее актуальной задачей теории расписаний для практических применений в календарном производственном планировании является задача оптимизации работы параллельно-последовательной обслуживающей системы (ППОС). В качестве ППОС может выступать любой производственный объект (цех или участок цеха, в целом предприятие или любое его структурное подразделение). Существует несколько подходов, применимых для точного решения этой задачи, однако большинство из них имеют, скорее, теоретическое, чем прикладное значение. Размерности и сложность порождаемых в рамках данных подходов моделей целочисленного программирования делают задачу оптимизации календарных планов ППОС неразрешимой даже для объектов уровня участка цеха с простейшими технологиями [1].

Предлагаемая работа развивает тему, ранее представленную автором на страницах журнала "Информационные технологии" [2], а также в работе [3].

1. Синтез расписаний параллельных динамических систем

В сравнении с работой [2] удалось несколько упростить постановку общей задачи оптимизации расписаний параллельных динамических обслуживающих систем (ПДОС). Компактное ее представление выглядит следующим образом.

Пусть известно расписание (значения задержек) поступления заявок в параллельную обслуживающую систему $T^0 = \|\tau_j^0\|$ и время обслуживания каждой заявки каждым прибором $T = \|t_{ij}\|$. Тогда задача оптимизации расписаний параллельной динамической ОС имеет вид:

$$\sum_{i=1}^{I} x_{ij} = 1,\ j = \overline{1, J}, \tag{1.1}$$

$$\underline{b}_j \leqslant \sum_{j=1}^{J} x_{ij} \leqslant \overline{b}_i,\ \ i = \overline{1, I}, \tag{1.2}$$

$$x_{ij} = \begin{cases} 1, \text{ если заявка } j \text{ назначается на прибор } i; \\ 0, \text{ в противном случае;} \end{cases} \tag{1.3}$$

$y_{ij} \geqslant 0$, переменная,
обнуляющая отрицательные задержки τ_{ij}:

$$\bar{\tau}_{ij} = \begin{cases} \tau_{ij}, \text{ если } \tau_{ij} \geqslant 0 \\ 0 \text{ в противном случае;} \end{cases} \tag{1.4}$$

$$\tau_{ij} = \tau_j^0 - \sum_{l=1}^{j-1} (\tau_{il} + t_{il}) x_{il},\ \forall i = \overline{1, I},\ j = \overline{1, J}; \tag{1.5}$$

$$\bar{\tau}_{ij} = \tau_{ij} + y_{ij} \geqslant 0,\ \forall i = \overline{1, I},\ j = \overline{1, J}; \tag{1.6}$$

$$\sum_{j=1}^{J} \bar{\tau}_{ij} x_{ij} + \sum_{j=1}^{J} t_{ij} x_{ij} \leqslant \lambda,\ i = \overline{1, I}; \tag{1.7}$$

$$Z_1 = \lambda \to \min. \tag{1.8}$$

Величина $\bar{\tau}_{ij}$ имеет смысл фактической задержки начала выполнения j-й заявки i-м прибором после завершения обслуживания им предшествующей заявки. Компенсирующие переменные y_{ij} вводятся для того, чтобы избежать появления отрицательных задержек τ_{ij}.

Выражения (1.7) и (1.8) реализуют минимаксный критерий, имеющий смысл критерия максимального быстродействия (иногда называемого также критерием равномерной загрузки) вида

$$Z_1 = \max_i \left\{ \sum_{j=1}^{J} (t_{ij} + \bar{\tau}_{ij}) x_{ij} \right\} \to \min.$$

Задача (1.1)—(1.8) содержит значения фактических задержек τ_{ij}, являющиеся рекурсивными функциями. Подробная процедура построения эквивалентной задачи линейного целочисленного программирования представлена в работе [2]. Удалось найти ее упрощение [4], опосредованное упрощением исходной задачи. В конечном итоге постановке (1.1)—(1.8) взаимно однозначно соответствует следующая задача частично-целочисленного программирования с булевыми переменными:

$$\sum_{i=1}^{I} x_{ij} = 1,\ \forall j = \overline{1, J}; \tag{1.9}$$

$$\underline{b}_i \leqslant \sum_{j=1}^{J} x_{ij} \leqslant \overline{b}_i,\ \forall i = \overline{1, I}; \quad (1.10)$$

$$x_{ij} = \begin{cases} 1, \text{ если заявка } j \text{ закрепляется} \\ \text{за прибором } i; \\ 0 \text{ в противном случае;} \end{cases} \quad (1.11)$$

$$u_{ijk} = \begin{cases} 1, \text{ при истинности выражения} \\ f_{ijk}(x_{ik}, x_{ik+1}, \ldots, x_{ij-1}) = \\ = x_{ij}x_{ik}\bigcap_{l=k+1}^{j-1} \overline{x_{il}},\ i = \overline{1, I},\ j = \overline{1, J}; \\ 0 \text{ в противном случае,} \end{cases} \quad (1.12)$$

где $\overline{x_{il}} = 1 - x_{il}$;

$$-K + 2 \leqslant x_{ij} + x_{ik} - \bigcap_{l=k+1}^{j-1} x_{il} - Ku_{ijk} \leqslant 1,$$

$$\forall i = \overline{1, I},\ j = \overline{1, J},\ k = \overline{1, J},\ K = j - k + 1; \quad (1.13)$$

$$\tilde{\tau}_{ij} = y_{ij} + \tau_j^0 - \sum_{k=1}^{j-1} (\tau_k^0 + t_{ik})u_{ijk} \geqslant 0, \text{ или}$$

$$-y_{ij} + \sum_{k=1}^{j-1} (\tau_k^0 + t_{ik})u_{ijk} \leqslant \tau_j^0,$$

$$i = \overline{1, I},\ j = \overline{1, J}; \quad (1.14)$$

$$\sum_{j=1}^{J} (\tau_j^0 + t_{ij})x_{ij} + \sum_{j=1}^{J} y_{ij} -$$

$$- \sum_{j=1}^{J} \sum_{k=1}^{j-1} (\tau_k^0 + t_{ik})u_{ijk} \leqslant \lambda,\ \forall i = \overline{1, I}; \quad (1.15)$$

$$y_{ij} \geqslant 0,\ \forall i = \overline{1, I},\ j = \overline{1, J}; \quad (1.16)$$

$$Z_1 = \lambda \to \min. \quad (1.17)$$

Размерность задач календарного планирования ПДОС (1.9)—(1.17) не позволяет непосредственно, используя стандартные алгоритмы целочисленной оптимизации, находить оптимальные расписания работы систем реальной размерности. Это влечет необходимость использования методов декомпозиции, точных и приближенных [5], а также упрощение самой задачи. В частности, экспериментально подтвержденную численную эффективность имеет подход, который приводит к следующей релаксированной постановке [2]:

$$\sum_{i=1}^{I} x_{ij} = 1,\ \forall j = \overline{1, J}; \quad (1.18)$$

$$\underline{b}_i \leqslant \sum_{j=1}^{J} x_{ij} \leqslant \overline{b}_i,\ \forall i = \overline{1, I}. \quad (1.19)$$

$$x_{ij} = \begin{cases} 1, \text{ если заявка } j \text{ назначается на прибор } i; \\ 0 \text{ в противном случае;} \end{cases} \quad (1.20)$$

$$\sum_{j=1}^{J} \tau_j^0 x_{ij} \leqslant \beta,\ \forall i = \overline{1, I}; \quad (1.21)$$

$$\sum_{j=1}^{J} t_{ij}x_{ij} \leqslant \lambda,\ \forall i = \overline{1, I}. \quad (1.22)$$

$$Z_2 = \lambda \to \min;\ Z_3 = \beta \to \min. \quad (1.23)$$

Линеаризуется векторный критерий $Z = \{Z_2, Z_3\}$ посредством свертки:

$$Z_4 = \alpha\lambda + (1 - \alpha)\beta, \quad (1.24)$$

где $0 \leqslant \alpha \leqslant 1$ — параметр модели. Заметим, что в (1.21) при больших значениях абсолютных задержек τ_j^0 корректнее использовать относительные задержки $\tilde{\tau}_j^0 = \tau_j^0 - \min_j (\tau_j^0),\ j = \overline{1, J}$.

2. Синтез расписаний многостадийных последовательных систем

В работе автора [3] представлен подход, позволяющий синтезировать близкие к оптимальным по быстродействию расписания многостадийных последовательных ОС за полиномиальное от размерности время.

Рассмотрим многостадийную обслуживающую систему, состоящую из k различных приборов, которая обслуживает L заявок. Маршруты обслуживания заявок приборами фиксированы, но различны. Длительности обслуживания каждой заявки каждым прибором также различны. Одновременно одним прибором может обслуживаться только одна заявка. Прерываний обслуживания не допускается. Альтернативные назначения отсутствуют (все назначения на параллельные приборы определены).

Пусть $M = \|\mu_{ls}\|$, $\mu_{ls}[\overline{1, k}]$, $T = \|t_{ls}\|$, $l = \overline{1, L}$, $s = \overline{1, S}$ — матрицы маршрутов и длительностей обслуживания L заявок на k различных приборах, при максимальном числе операций над одной заявкой S. Далее удобно перейти от двойной индексации операций обслуживания заявок приборами к одинарной ($ls = i$), учитывая уникальность каждой операции. Поэтому длительность i-й операции обозначим t_i. Введем также обозначения:

A — множество операций обслуживания заявок;

U — множество логических условий предшествования-следования;

V — множество логических условий неодновременности выполнения операций;

$G = (A, U, V)$ — некоторый оператор над множествами операций и логических условий, которым должен удовлетворять процесс обслуживания. В частности, $G = (A, U, V)$ можно интерпретировать как смешанный граф с множеством вершин A, множеством дуг U и множеством ребер V.

$I_G = I_{(A, U, V)}$ — множество индексов операций и логических условий, определяемых на G;

n — общее число операций;

ξ_i — искомое время начала выполнения операции i, $\forall i \in I_{(A, \varnothing, \varnothing)}$.

Целесообразно использовать фиктивные начальную и конечную операции с временем начала выполнения соответственно 0 и ξ_n. Тогда задача календарного планирования может быть представлена следующим образом:

$$Z = \xi_n \to \min, \qquad (2.1)$$

$$\xi_i - \xi_j + t_i \leqslant 0, \quad \forall (i, j) \in I_{(A, U, \varnothing)}, \qquad (2.2)$$

если известно, что операция j непосредственно следует во времени за операцией i.

Если операции i и j одновременно не могут выполняться, но последовательность их выполнения заранее не оговорена, то должно быть справедливо одно из неравенств: либо $\xi_j - \xi_i - t_i \geqslant 0$, либо $\xi_i - \xi_j - t_j \geqslant 0$. Поставим в соответствие каждой паре таких операций i и j величину w_{ij}, принимающую значение $w_{ij} = 1$, если операцию i решено выполнять раньше операции j, и $w_{ij} = 0$ в противном случае. Тогда ограничение на порядок следования операций можно записать в следующем виде:

$$Bw_{ij} - t_j \geqslant \xi_j - \xi_i \geqslant t_i - B(1 - w_{ij}),$$
$$B > t_j, \; B > t_i, \; \forall (i, j) \in I_{(A, \varnothing, V)}, \qquad (2.3)$$

$$w_{ij} = \begin{cases} 1, \text{ если операция } i \\ \text{предшествует } j, \\ 0, \text{ если операция } j \\ \text{предшествует } i, \end{cases} \quad \forall (i, j) \in I_{(A, \varnothing, V)}, \qquad (2.4)$$

$$\xi_i \geqslant 0, \quad i = \overline{1, n}. \qquad (2.5)$$

С учетом ограничений (2.2)—(2.5) возникает проблема выбора календарного плана, оптимального либо по критерию быстродействия (2.1), включая разновидности, либо по критерию стоимости. Данная формальная постановка задачи календарного производственного планирования является одной из самых компактных [1]. Существует ряд альтернативных постановок [1, 3], однако большинство из них порождают задачи целочисленного программирования гигантских размерностей при описании даже сравнительно простых объектов [1, 6].

Поставим в соответствие (2.1)—(2.5) сетевую модель в представлении "узел—операция". Отдельные операции отображаются вершинами (узлами) сети, с помощью дуг задаются технологические маршруты обслуживания заявок, с помощью ребер — условия неодновременности обслуживания различных заявок одним прибором. Представленная на рис. 1 сеть отображает определенный выше смешанный граф $G = (A, U, V)$.

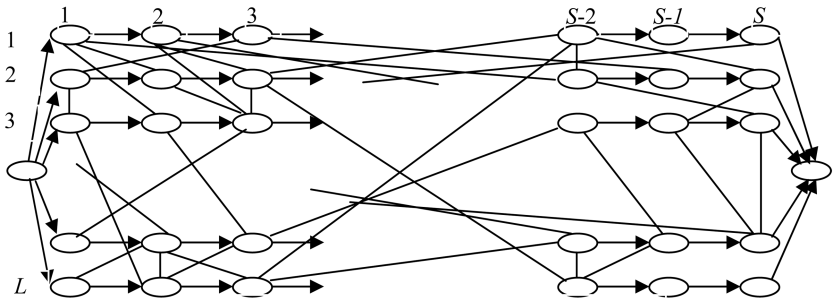


Рис. 1. Смешанный граф $G = (A, U, V)$

При замене произвольного ребра дугой устанавливается отношение предшествования — следования между двумя операциями. Получаемая в результате замены всех ребер дугами нормальная бесконтурная сеть определяет допустимое расписание работы ОС.

Поэтому оптимальное решение сформулированной выше задачи (2.1)—(2.5) графически эквивалентно такой замене всех ребер смешанной сети G дугами, которая определяет нормальную бесконтурную сеть, порождающую расписание минимальной длины. Очевидно, что верхняя оценка Λ числа вариантов решения рассматриваемой задачи календарного планирования равна 2^K. Где $K = \text{argmax}(I_{(\varnothing, \varnothing, V)})$ — число ребер в сети G. В работе [3] приведены точный и приближенный (полиномиально трудоемкий) алгоритмы решения этой задачи.

3. Синтез расписаний параллельно-последовательных систем

Представленная ниже формальная постановка общей задачи синтеза расписаний ППОС является комбинацией и обобщением задач (1.1)—(1.8) и (2.1)—(2.5).

Далее используем следующие обозначения:

p — номер подсистемы (многоканального прибора), $p = \overline{1, n}$;

j — номер заявки, J — число заявок, $j = \overline{1, J}$;

I_p — множество взаимозаменяемых приборов в подсистеме p, $p = \overline{1, n}$;

i — номер прибора (канала многоканального прибора), $i \in I_p$, $p = \overline{1, n}$;

q — номер этапа динамической модели $q = 1, 2, \ldots$;

s — номер обращения заявки j к подсистеме p в соответствии с технологическим маршрутом, $s = 1, 2, \ldots$;

t_{jpis} — время обслуживания j-й заявки i-м прибором p-й подсистемы при s-м обращении;

t_{jpi} — то же при последнем обращении;

t_{ji} — то же для замыкающей подсистемы;

c_{jpis} — оценка качества обслуживания j-й заявки i-м прибором p-й подсистемы при s-м обращении;

$M = \|p_{js}\|$ — матрица технологических маршрутов; определена с точностью до блоков (подсистем) ППОС;

$$x^q_{jpis} = \begin{cases} 1, \text{ если заявка } j \text{ закрепляется} \\ \text{за прибором } i \text{ подсистемы } p \\ \text{при обращении } s \text{ на шаге } q, \\ 0 \text{ в противном случае;} \end{cases} \quad (3.1)$$

x^q_{jpi} — то же при последнем обращении;

x^q_{ji} — то же для замыкающего блока;

τ^{q-1}_{jps} — расписание поступления заявок j на вход подсистемы p на шаге q, при обращении s,

τ^{q-1}_{jp} — то же при последнем обращении, τ^{q-1}_{j} — то же для последней подсистемы маршрута;

τ^q_{jps} — расписание обслуживания (поступления заявок j на выход) подсистемы p на шаге q при обращении s, τ^q_{jp} — то же при последнем обращении, τ^q_j — то же для последней подсистемы маршрута;

x^q_{jpis} — фактическая задержка начала выполнения прибором i подсистемы p заявки j при обращении s после завершения обслуживания им предшествующей заявки, τ^q_{jpi} — то же при последнем обращении, τ^q_{ji} — то же для последней подсистемы маршрута.

Расписания τ^{q-1}_{jps}, τ^q_{jps} и τ^q_{jpis} связаны соотношением:

$$\tau^q_{jps} = \tau^{q-1}_{jps} + \sum_{i \in I_p} \tau^q_{jpis} x^q_{jpis}.$$

Тогда общая задача оптимизации расписаний ППОС представляется следующим образом:

$$\sum_{i \in I_p} x^q_{jpis} = 1, j = \overline{1, J},$$

$$q = 1, 2, ..., s = 1, 2, ..., p = \overline{1, n}; \quad (3.2)$$

$$\underline{b}^q_{pis} \leqslant \sum_{j=1}^{J} x^q_{jpis} \leqslant \bar{b}^q_{pis}, \forall i \in I_p,$$

$$q = 1, 2, ..., s = 1, 2, ..., p = \overline{1, n}; \quad (3.3)$$

$$y^q_{jpis} \geqslant 0, \forall i \in I_p, q = 1, 2, ..., s = 1, 2, ...,$$

$$p = \overline{1, n}, j = \overline{1, J} \quad (3.4)$$

(переменные, компенсирующие возможные отрицательные значения задержек τ^q_{ijps});

$$\tilde{\tau}^q_{jpis} = \begin{cases} \tau^q_{jpis}, \text{ если } \tau^q_{jpis} \geqslant 0, \\ \forall i \in I_p, q = 1, 2, ..., s = 1, 2, ..., \\ p = \overline{1, n}, j = \overline{1, J}; \\ 0 \text{ в противном случае,} \end{cases} \quad (3.5)$$

$$\tau^q_{jpis} = \tau^{q-1}_{jp} - \sum_{l=1}^{j-1} (\tau^q_{lpis} + t_{lpis}) x^q_{lpis},$$

$$\forall i \in I_p, q = 1, 2, ..., s = 1, 2, ...,$$

$$p = \overline{1, n}, j = \overline{1, J}; \quad (3.6)$$

$$\tilde{\tau}^q_{jpis} = \tau^q_{jpis} + y^q_{jpis} \geqslant 0, \forall i \in I_p, q = 1, 2, ...,$$

$$s = 1, 2, ..., p = \overline{1, n}, j = \overline{1, J}; \quad (3.7)$$

$$\tau^q_{jps} = \sum_{i \in I_p} (\tilde{\tau}^q_{jpis} + t_{jpis}) x^q_{jpis},$$

$$\forall q = 1, 2, ..., s = 1, 2, ..., p = \overline{1, n}, j = \overline{1, J}; \quad (3.8)$$

$$\sum_{j=1}^{J} \tilde{\tau}^q_{ji} x^q_{ji} + \sum_{j=1}^{J} t_{ji} x^q_{ji} \leqslant \lambda, \forall i = \overline{1, I}; \quad (3.9)$$

$$\tilde{\tau}^q_{jpis} - \tilde{\tau}^q_{jp'i's'} + t_{jpis} x^q_{jpis} \leqslant 0,$$

$$\forall (jpis, jp'i's') \in I_{(A, U, \varnothing)}; \quad (3.10)$$

$$Bw_{jpis, j'pis'} - t_{j'pis'} x^q_{j'pis'} \geqslant \tilde{\tau}^q_{j'pis'} - \tilde{\tau}^q_{jpis} \geqslant t_{jpis} x^q_{jpis} - B(1 - w_{jpis, j'pis'}), \forall (jpis, j'pis') \in I_{(A, \varnothing, V)}; \quad (3.11)$$

$$w_{jpis, j'pis'} = \begin{cases} 1, \text{ если операция } jpis \text{ предшествует } j'pis'; \\ 0, \text{ если операция } j'pis' \text{ предшествует } jpis, \end{cases}$$

$$\forall (jpis, j'pis') \in I_{(A, \varnothing, V)}; \quad (3.12)$$

$$Z_3 = \lambda \to \min. \quad (3.13)$$

Данная задача содержит рекурсивные функции (3.6)—(3.8), формально идентичные (1.5). Нет сомнений относительно возможности их раскрытия и построения задачи ЛП с булевыми переменными, подобной задаче (1.9)—(1.17). Также нет сомнений в отсутствии практической целесообразности построения такой модели. Действительно, ее размерность многократно превысит размерность задачи (1.9)—(1.17), ранее признанной неразрешимой для объектов с реальными характеристиками размерности [2]. Поэтому определение оптимальных расписаний посредством прямого применения глобальной модели ППОС (3.1)—(3.13) пока невозможно даже теоретически.

Для корректного решения практических задач календарного планирования приходится строить итерационные алгоритмы, основанные на разбиении глобальной задачи оптимизации расписаний ППОС на подзадачи, с оптимизацией расписаний локальных подзадач и итерационной синхронизацией их входных и выходных параметров. Подобный алгоритм неполной декомпозиции представлен ниже.

Определение следующих входных параметров алгоритма: 1) способа разбиения множества операций A с маршрутами $M = \|p_{js}\|$ на подмножества A^q (и образованием подграфов $G^q = (A^q, U^q, V^q)$); 2) способа перехода от G^q к G^{q+1} (и A^q к A^{q+1}).

Процедура формирования подграфов $G^q = (A^q, U^q, V^q)$ во многом зависит от используемых вычислительных схем оптимизации расписаний на каждом шаге q.

Поскольку на любом шаге q для оптимизации назначений на параллельные приборы x^q_{jpis}, $i \in I_p$, $p = \overline{1, n}$ предпочтительна модель (1.18)—(1.24), которая не учитывает назначений на предшествующих шагах, все x^q_{jpis} необходимо фиксировать и учитывать далее в начальных задержках τ^{q-1}_{jps}.

Алгоритм А1.

1. На предварительном шаге ($q = 1$) в соответствии с принципом равномерности потоков операций относительно заявок определим величину m_1 — число столбцов матрицы маршрутов $M = \|p_{js}\|$, $s = \overline{1, m_1}$, на основе которых формируется A^1. Величину m_1 для краткости будем именовать толщиной слоя A^1. Не теряя общности, можно принять $m_q = m_1$ (зафиксировать для всех шагов), тогда m_1 — толщина слоя A^q. Значение это определяется максимальным числом конфликтов по приборам в слое A^q, которое должно находиться в интервале [1 ÷ 100]. Настоящее ограничение в дальнейшем обеспечивает полиномиальную трудоемкость алгоритма. В свою очередь, для определения числа конфликтов по приборам в начальном слое A^1 необходимо найти назначения x^1_{jpis}, $i \in I_p$, $p = \overline{1, n}$, $s = \overline{1, m_1}$. A и A^q можно представить в виде матриц $\tilde{A} = \|\tilde{a}_{js}\|$ и $\tilde{A}^q = \|\tilde{a}^q_{js}\|$, $s = \overline{1, m_q}$, $j = \overline{1, J}$. Элементы $\tilde{a}_{js}$ отображаются парой чисел $\tilde{a}_{js} = (p_{js}, t_{js})$. Если маршруты $M = \|p_{js}\|$ различаются числом операций, то часть недостающих элементов t_{js}, p_{js} и $\tilde{a}_{js}$ для малооперационных маршрутов дополняется нулями. Соотношения для $\tilde{A}^q$: $\tilde{a}^q_{js} = \tilde{a}^j_{js}$, $s = \overline{1, m_q}$, $s_j = \overline{q, q + m_q}$, $j = \overline{1, J}$.

Последовательное решение подзадач (1.18)—(1.24) для всех параллельных подсистем (если таковые имеют место) позволяет определить локальные назначения x^1_{ijps} для слоя A^1. Фиксируем назначения x^1_{ijps}. Тогда преобразованный слой A^1 будет содержать только последовательно выполняющиеся операции. Поэтому A^1 однозначно определяет подграф $G^1 = (A^1, U^1, V^1)$ и подзадачу (2.1)—(2.5). Для ее решения применим модификацию алгоритма ветвей и границ [3]. В результате получим расписание $\bar{\tau}^1_{jpis}$, локально оптимальное для слоя A^1. Начальные задержки τ^0_{jpis} можно учесть, введя дополнительные (начальные) бесконфликтные операции либо веса дуг U^1. В последнем случае веса всех дуг, кроме начальных, равны нулю. Для получения более точной оценки длины расписания на этом шаге можно рассмотреть граф $\bar{G}^1 = (A, U, V^1)$, в котором вершины, принадлежащие множеству $A \backslash A^1$, соединены только дугами из $U \backslash U^1$.

2. $q := q + 1$. Переход от слоя A^q к A^{q+1} состоит: а) в исключении элементов $\tilde{a}^q_{j1}$ из $\tilde{A}^q$, при этом $\tilde{a}^{q+1}_{js-1} = \tilde{a}^q_{js}$, $s = \overline{2, m_q}$; б) в дополнении $\tilde{A}^q$ следующими по порядку элементами $\tilde{a}^{q+1}_{js}$, $s = m_q + 1$, если для маршрута j добавляемый элемент не приводит к увеличению оценки длины расписания. В противном случае строка j в $\tilde{A}^{q+1}$ эквивалентна $\tilde{A}^q$ ($\tilde{a}^{q+1}_{js} = \tilde{a}^q_{js}$, $s = \overline{1, m_q}$). Особые (наилучшие) условия возникают тогда, когда добавляемые вершины приводят к увеличению оценок длины расписания для всех маршрутов j одновременно. При этих обстоятельствах исключаются все начальные вершины A^q (первый столбец из $\tilde{A}^q$) ($\tilde{a}^{q+1}_{js-1} = \tilde{a}^q_{js}$, $s = \overline{2, m_q}$, $j = \overline{1, J}$) и в качестве последнего столбца добавляются очередные вершины из A: $\tilde{a}^{q+1}_{js}$, $s = m_q + 1$, $j = \overline{1, J}$.

3. Для слоя A^q определяем предшествующие и последующие задержки обслуживания по каждой из заявок $j = \overline{1, J}$ для всех операций, не вошедших в слой. Предшествующие назначения x^l_{ijps} на шагах $l = \overline{1, q-1}$ сохраняются. Предшествующие задержки определяются посредством МКП, поскольку для соответствующих операций конфликты разрешены на шагах $1 \div (q - 1)$. Оценки последующих задержек также определяются посредством МКП для не вошедших в слой последующих операций.

4. Определяются оптимальные, в смысле и посредством решения задачи (1.18)—(1.24) назначения x^q_{ijps} для вновь включенных в слой A^q вершин.

5. Поскольку в текущем слое A^q определены все назначения x^q_{ijps} и таким образом слой преобразуется в m_q-стадийную последовательную обслуживающую систему, определяем его локально оптимальное расписание $\bar{\tau}^q_{jpis}$ посредством модификации алгоритма ветвей и границ [3].

6. Проверка условий останова алгоритма. Если текущий слой A^q является последним $\tilde{a}^{q+1}_{js} = \tilde{a}^q_{js}$, $s = \overline{1, m_q}$, $j = \overline{1, J}$, получено локально оптимальное расписание ППОС, иначе следует перейти к п. 2.

Рассмотрим иллюстративный пример.

Таблица 1

Технологические маршруты

Номер заявки	Номер многоканального прибора (подсистемы)			
1	1	2	1	3
2	3	2	1	2
3	1	2	1	0
4	2	1	2	0
5	2	3	1	2
6	3	1	2	1
7	1	3	2	1
8	2	1	3	1
9	2	1	2	1
10	1	2	3	2

Таблица 2

Время обслуживания ППОС t_{jpis}

Номер заявки	Время обслуживания заявок каналами приборов								Начальная задержка
	1	2	1	2	1	2	1	2	
1	2	3	1	2	3	2	1		0
2	3		2	1	2	2	1	3	0
3	3	2	3	4	3	5	0		0
4	1	3	2	3	3	2	0		1
5	2	3	3		3	2	3	2	2
6	2		2	4	4	3	1	1	2
7	4	3	2		1	2	2	1	3
8	2	2	3	2	3		2	4	3
9	1	4	2	1	2	2	1	2	4
10	2	3	4	1	2		1	3	5

В табл. 1 и табл. 2 приведены данные о ППОС, состоящей из пяти приборов (каналов) трех типов (по два неидентичных прибора (канала) первого и второго типов и один прибор третьего типа). Системе предстоит обслужить 10 заявок в соответствии с заданными последовательностями. Технологические маршруты отображены в табл. 1, время обслуживания заявок каналами приборов — в табл. 2. Для удобства клетки таблиц, содержащие данные о каналах прибора первого типа, выделены темной заливкой, третьего типа — светлой заливкой, второго типа заливкой не выделены.

Для решения задачи синтеза оптимального расписания алгоритм А1 применен со следующими параметрами. Толщина слоя $m_q = 2$ постоянна для всех шагов q. Сделанные на предшествующих шагах назначения заявок на каналы приборов далее не изменяются.

При данных параметрах локально оптимальное решение было найдено на шаге 4 алгоритма. В результате определены назначения и расписания обслуживания заявок на каждом шаге q (табл. 3). Таким образом, в процессе расчетов были сгенерированы и решены восемь подзадач (1.18)—(1.24) и, соответственно, три подзадачи (2.1)—(2.5). Получено локально оптимальное расписание, представленное в виде графика загрузки приборов (и отдельных каналов) на рис. 2. На поверку оно совпало с оптимальным по быстродействию расписанием. В этом нетрудно непосредственно убедиться даже при поверхностном анализе полученных результатов.

В табл. 3 заливкой выделены абсолютные задержки (расписания) на входе параллельных подсистем (многоканальных приборов), получаемые на каждом шаге алгоритма А1, а также относительные (нормализованные) задержки, используемые в критерии эффективности подзадач (1.18)—(1.24).

Оценим эффективность алгоритма. Алгоритм А1 сводит синтез расписаний ППОС к решению огра-

Таблица 3

Локально оптимальные решения

p	Прибор 1 ($p = 1$)					Прибор 2 ($p = 2$)				
Шаг	Заявка	Задержка на входе абс/отн		Назначение x^q_{j1is} каналы $p.i$		Заявка	Задержка на входе абс/отн		Назначение x^q_{j2is} каналы $p.i$	
				1.1	1.2				2.1	2.2
q	j	τ^{q-1}_{jps}	τ^{q-1}_{jps}	$i = 1$	$i = 2$	j	τ^{q-1}_{jps}	τ^{q-1}_{jps}	$i = 1$	$i = 2$
$q = 1$	1	0	0	1	0	4	1	0	1	0
	3	0	0	0	1	5	2	1	0	1
	7	3	3	0	1	8	3	2	0	1
	10	5	5	1	0	9	4	3	1	0
$q = 2$	4	2	0	1	0	1	2	0	1	0
	6	6	4	0	1	2	3	1	1	0
	8	7	5	1	0	3	2	0	1	0
	9	5	3	0	1	10	7	5	0	1
$q = 3$	1	10	5	0	1	4	4	0	1	0
	2	5	0	0	1	6	11	7	0	1
	3	8	3	1	0	7	10	6	0	1
	5	8	3	1	0	9	12	8	1	0
$q = 4$	6	15	3	1	0	2	16	3	1	0
	7	12	0	0	1	5	13	0	0	1
	8	13	1	1	0	10	15	2	1	0
	9	15	3	0	1	0	0	0	0	0

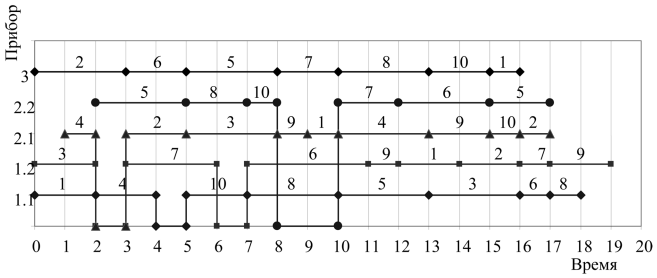


Рис. 2. График загрузки приборов

ниченной последовательности задач (1.18)—(1.24) и (2.1)—(2.5). Численные эксперименты показывают, что для получения оптимальных по быстродействию расписаний ППОС не обязательно нахождение оптимальных назначений в рамках задач (1.9)—(1.17) либо (1.18)—(1.24). Свойства алгоритма таковы, что не лучшие по быстродействию локальные назначения компенсируются алгоритмом при решении подзадач (2.1)—(2.5). Таким образом, ППОС в подавляющем большинстве случаев присуща множественность оптимальных расписаний. И алгоритм A1 находит одно из оптимальных расписаний. Поэтому трудоемкость алгоритма оптимизации расписаний ППОС главным образом зависит от трудоемкости алгоритмов решения задач линейного программирования с булевыми переменными (1.18)—(1.24) и (2.1)—(2.5). Поскольку размерности всех подзадач, порождаемых алгоритмом A1, соответствующим образом ограничены, фактически A1 имеет полиномиальную трудоемкость.

Заключение

Проведенные численные эксперименты с представленными выше моделями и алгоритмом позволяют сделать вывод о наличии условий, при которых алгоритм A1 гарантирует получение оптимальных по быстродействию расписаний ППОС. Потенциальные размерности NP-трудных подзадач (1.18)—(1.24) оптимизации расписаний параллельных ОС для реальных производственных объектов невелики. Число элементов (каналов), как правило, находится в интервале [2...50], поэтому решение подзадач (1.18)—(1.24) труда не составляет. А поскольку алгоритм [3] синтеза субоптимальных расписаний последовательных ОС содержит полиномиальное от размерности число шагов, то и алгоритм A1 фактически является полиномиальным, что можно считать основным достижением проведенного исследования.

Работа выполнена при поддержке гранта РГНФ 2009 г. 09-02-00093 а/И.

Список литературы

1. **Танаев В. С., Шкурба В. В.** Введение в теорию расписаний. М.: Наука, 1975. 256 с.

2. **Мезенцев Ю. А.** Оптимизация расписаний параллельных динамических систем в календарном планировании // Информационные технологии. 2008. № 2. С. 16—23.

3. **Мезенцев Ю. А.** Алгоритмы синтеза расписаний многостадийных обслуживающих систем в календарном планировании // Омский научный вестник. 2006. № 3 (36) С. 97—102.

4. **Иванов Л. Н., Мезенцев Ю. А.** Методы оптимизации расписаний параллельных обслуживающих систем // Программные продукты и системы. // 2008. № 1. С. 72—74.

5. **Мезенцев Ю. А.** Декомпозиционный метод решения одного класса задач оптимального проектирования // Научный вестник НГТУ. 2006. № 3 (24). С. 67—100.

6. **Танаев В. С., Сотсков Ю. Н., Струсевич В. А.** Теория расписаний. Многостадийные системы. М.: Наука, 1989. 328 с.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

УДК 004.052.3

В. А. Богатырев, д-р техн. наук, проф.,
С. В. Богатырев, аспирант,
Санкт-Петербургский государственный университет информационных технологий механики и оптики,
e-mail: bva@tinuviel.ru

Объединение резервированных серверов в кластеры высоконадежной компьютерной системы

Поставлена и решена задача оптимизации объединения в кластеры резервированных серверов различного функционального назначения. Показано, что при многоуровневой конфигурации коммуникационной подсистемы для обеспечения высокой отказоустойчивости и надежности целесообразно объединение в кластеры разнотипных по функциональности серверов.

Ключевые слова: *надежность, отказоустойчивость, многоуровневая компьютерная система, векторная оптимизация, кластер.*

Введение

К компьютерным системам предъявляются требования высокой надежности, отказоустойчивости и производительности при их низкой стоимости. Современные высоконадежные компьютерные системы, в том числе центры обработки данных, как правило, строятся на основе сетевых технологий и включают в свой состав резервированные серверы различного функционального назначения (Web-серверы, почтовые серверы, серверы баз данных, FTP-серверы и т. п.). Подключение серверов в систему осуществляется через многоуровневую коммуникационную подсистему, при этом надежность и эффективность системы зависит не только от кратности резервирования серверов, но и от вариантов их объединения в группы (кластеры), в частности, от включения в каждый кластер однотипных либо разнотипных серверов. Исследование влияния вариантов объединения (группирования) серверов в кластеры рассматривается в предлагаемой статье.

Постановка задачи

Рассматривается многоуровневая компьютерная система, в которой за основу построения коммуникационной подсистемы взята древовидная топологии, в простейшем случае с выделением коммутационных узлов верхнего и нижнего уровней. Для обеспечения высокой надежности и производительности серверы резервируются и объединяются в группы — кластеры. Каждая группа серверов (кластер) подключается к сети через отдельный коммутатор (коммутационный узел) нижнего уровня (КНУ). Все КНУ взаимосвязаны через корневой коммутатор (коммутационный узел) верхнего уровня (КВУ). Для повышения надежности и отказоустойчивости КВУ и КНУ могут резервироваться.

В системе выделяются n типов серверов, различающихся по функциональному назначению. Рассматриваются варианты объединения в каждый кластер как однотипных, так и разнотипных по функциональности серверов.

Будем считать, что условие работоспособности системы заключается в работоспособности и доступности (связанности) хотя бы одного сервера каждого функционального типа, вне зависимости от его расположения. Требуется по критерию достижения максимальной надежности определить наиболее рациональный вариант объединения серверов в группы (кластеры) и для него оптимизировать распределение кратности резервирования узлов различных типов с учетом достигаемого уровня надежности системы и затрат на ее реализацию.

Структура исследуемой многоуровневой компьютерной системы с резервированием коммутационных узлов верхнего и нижнего уровней (КВУ, КМУ), а также серверов, объединяемых в кластеры, приведена на рис. 1.

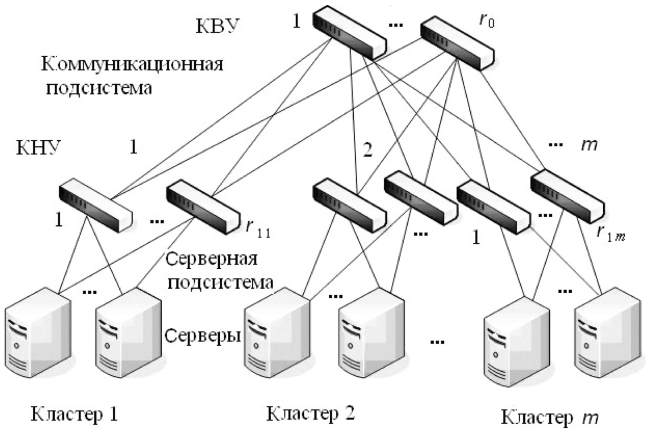


Рис. 1. Структура исследуемой многоуровневой компьютерной системы

Варианты объединения серверов в кластеры

В компьютерных системах в зависимости от их назначения возможно использование как однотипных, так и разнотипных по функциональному назначению серверов. Однотипные по функциональному назначению серверы могут быть как идентичными, так и различными по техническим характеристикам (параметрам).

В компьютерной системе серверы могут объединяться в кластеры, причем в системе может быть организовано несколько кластеров. При неоднородности серверов их группирование (объединение) в кластеры в значительной мере влияет на эффективность системы.

Рассматриваемые варианты объединения серверов в кластеры представлены на рис. 2.

Рассмотрим объединение серверов в кластеры, когда в компьютерных системах выделяются n типов резервированных серверов, различающихся по функциональному назначению. Выделяются два основных подхода к объединению серверов в кластеры. При первом подходе в каждый кластер включаются однотипные по функциональному назначению серверы (однофункциональные кластеры), в этом случае при наличии n функциональных типов серверов в первый кластер включаются все серверы первого функционального типа, во второй кластер — все серверы второго типа и, наконец, в n-й кластер объединяются все серверы n-го функционального типа. Таким образом, организуется нескольких групп серверов (кластеров), каждый из которых предназначен для решения определенной задачи. Если число серверов каждого функционального типа значительно, то возможна организация нескольких кластеров ка-

Рис. 2. Варианты объединения серверов в кластеры

ждого функционального типа (т. е. происходит резервирование как серверов внутри кластеров, так и самих кластеров в целом).

При втором подходе объединения серверов в кластеры в каждый кластер включаются функционально разнотипные серверы (многофункциональные кластеры). При этом возможен случай, когда кластеры составляются из функционально разнородных серверов всех n типов (полнофункциональный кластер) и когда число N типов серверов в каждой группе (кластере) меньше общего числа их типов в системе, т. е. $N < n$ (не полнофункциональный кластер), причем возможно, что $N < n$ или $N > n$.

При выделении n типов серверов по функциональному назначению вариант объединения серверов в m кластеров охарактеризуем матрицей $\|\varphi_{ij}\|_{n \times m}$, элемент которой φ_{ij} равен числу исправных серверов i-го типа в j-м кластере.

Если сервер i-го типа включен в j-й кластер с кратностью резервирования $r/1$, то в исходном состоянии (до отказов) $\varphi_{ij} = r$; если сервер i-го типа в j-й кластер не включен или все его экземпляры отказали, то $\varphi_{ij} = 0$.

Случай комплектования каждого из m кластеров полным набором функциональных типов серверов (полнофункциональный кластер) по одному серверу каждого типа, когда $N = n$, где N — число типов функционально различных серверов в кластере ($i = 1, 2, ..., m$), представляется матрицей $\|\varphi_{ij}\|_{n \times m}$, все элементы которой $\varphi_{ij} = 1$. В частности, при $n = m = N = 4$ матрица $\|\varphi_{ij}\|_{n \times m}$ имеет вид

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \qquad (1)$$

Случай комплектования каждого кластера неполным набором функциональных типов серверов (неполнофункциональный кластер), когда в каждом кластере объединяются по $N < n$ типов функциональных серверов, если $N \leqslant n/2$, представляется матрицей $\|\varphi_{ij}\|_{n \times m}$ вида

$$\begin{bmatrix} E_1 & 0 & 0 & \dots & 0 \\ 0 & E_2 & 0 & \dots & 0 \\ 0 & 0 & E_3 & \dots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \dots & E_z \end{bmatrix}, \qquad (2)$$

где E_i — подматрицы, содержащие все единичные элементы, остальные элементы матрицы, не принадлежащие E_i, равны 0;

$$\varphi_{ij} = \begin{cases} 1, \text{ если } \varphi_{ij} \in E, \\ 0, \text{ если } \varphi_{ij} \notin E. \end{cases}$$

Число одинаково укомплектованных кластеров при кратности резервирования серверов r определяется как m/r (матрица (2) представляет случай, когда m/r — целое).

Случаю неполнофункциональности кластеров при $n > N > n/2$ соответствует матрица $\|\varphi_{ij}\|_{n \times m}$ вида

$$\begin{bmatrix} O_1 & E & E & \dots & E \\ E & O_2 & E & \dots & E \\ E & E & O_3 & \dots & E \\ E & E & E & \ddots & E \\ E & E & E & \dots & O_z \end{bmatrix}, \qquad (3)$$

где O_i — подматрицы, содержащие все нулевые элементы; элементы матрицы, не принадлежащие O_i, равны 1;

$$\varphi_{ij} = \begin{cases} 1, \text{ если } \varphi_{ij} \notin O; \\ 0, \text{ если } \varphi_{ij} \in O. \end{cases}$$

Таким образом, в матрице (3) все элементы, принадлежащие подматрицам E, равны единицам.

Например, при $n = 8$, $N = 4$, $m = 8$, $r = 4$ матрица (2) представима в виде

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}. \qquad (4)$$

При этом же числе кластеров и серверов возможны другие варианты комплектации кластеров, например

$$\begin{bmatrix} 1&1&1&1&0&0&0&0\\ 0&1&1&1&1&0&0&0\\ 0&0&1&1&1&1&0&0\\ 0&0&0&1&1&1&1&0\\ 0&0&0&0&1&1&1&1\\ 1&0&0&0&0&1&1&1\\ 1&1&0&0&0&0&1&1\\ 1&1&1&0&0&0&0&1 \end{bmatrix} \text{ или } \begin{bmatrix} 1&0&0&0&0&1&1&1\\ 1&1&0&0&0&0&1&1\\ 1&1&1&0&0&0&0&1\\ 1&1&1&1&0&0&0&0\\ 0&1&1&1&1&0&0&0\\ 0&0&1&1&1&1&0&0\\ 0&0&0&1&1&1&1&0\\ 0&0&0&0&1&1&1&1 \end{bmatrix}. \quad (5)$$

При кратности резервирования серверов $r > m/2$ для $n = 8$, $N = 6$, $m = 8$, $r = 6$ матрица (2) имеет вид

$$\begin{bmatrix} 0&0&1&1&1&1&1&1\\ 0&0&1&1&1&1&1&1\\ 1&1&0&0&1&1&1&1\\ 1&1&0&0&1&1&1&1\\ 1&1&1&1&0&0&1&1\\ 1&1&1&1&0&0&1&1\\ 1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&0&0 \end{bmatrix}. \quad (6)$$

При этом же числе серверов возможны другие варианты комплектации кластеров, например

$$\begin{bmatrix} 0&0&1&1&1&1&1&1\\ 1&0&0&1&1&1&1&1\\ 1&1&0&0&1&1&1&1\\ 1&1&1&0&0&1&1&1\\ 1&1&1&1&0&0&1&1\\ 1&1&1&1&1&0&0&1\\ 1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&0 \end{bmatrix} \text{ или } \begin{bmatrix} 0&1&1&1&1&1&1&0\\ 0&0&1&1&1&1&1&1\\ 1&0&0&1&1&1&1&1\\ 1&1&0&0&1&1&1&1\\ 1&1&1&0&0&1&1&1\\ 1&1&1&1&0&0&1&1\\ 1&1&1&1&1&0&0&1\\ 1&1&1&1&1&1&0&0 \end{bmatrix}. \quad (7)$$

В общем виде распределение r-резервируемых серверов разных функциональных типов по m кластерам, когда $r \leqslant m/2$, представим фрагментом матрицы $\|\varphi_{ij}\|_{n \times m}$ (рис. 3).

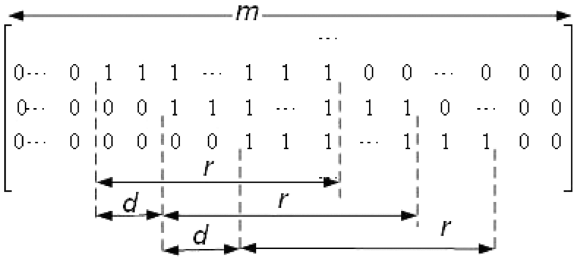


Рис. 3. Фрагмент матрицы распределение r-резервируемых серверов разных функциональных типов по кластерам

При этом d — смещение по числу кластеров размещения серверов i-го и $(i + 1)$-го функционального типа.

Заметим, что при $d = 1$ и $m = n$ формируются варианты, представленные матрицами (5), (7), а при $d = r$ — матрицами (2), (4), (6); если $r = m$, то матрица содержит все единичные элементы.

При $m = n = 8$, $d = r$ для $r = 4$, $d = 4$ и $r = 2$, $d = 2$ распределение серверов по кластерам представляется соответственно матрицами

$$\begin{bmatrix} 1&1&1&1&0&0&0&0\\ 0&0&0&0&1&1&1&1\\ 1&1&1&1&0&0&0&0\\ 0&0&0&0&1&1&1&1\\ 1&1&1&1&0&0&0&0\\ 0&0&0&0&1&1&1&1\\ 1&1&1&1&0&0&0&0\\ 0&0&0&0&1&1&1&1 \end{bmatrix}, \begin{bmatrix} 1&1&0&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&1\\ 1&1&0&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&1&1 \end{bmatrix},$$

которые с учетом инвариантности системы к порядку нумерации функциональных типов серверов (т. е. инвариантности матрицы к перестановке строк) приводятся к матрицам вида (4) и (6).

В случае объединения в кластеры однотипных по функциям серверов при том же суммарном составе серверов в системе, что и для вариантов (1), (2) и (3), имеем соответственно структуры, представляемые матрицами:

$$\begin{bmatrix} 4&0&0&0\\ 0&4&0&0\\ 0&0&4&0\\ 0&0&0&4 \end{bmatrix}, \begin{bmatrix} 4&0&0&0&0&0&0&0\\ 0&4&0&0&0&0&0&0\\ 0&0&4&0&0&0&0&0\\ 0&0&0&4&0&0&0&0\\ 0&0&0&0&4&0&0&0\\ 0&0&0&0&0&4&0&0\\ 0&0&0&0&0&0&4&0\\ 0&0&0&0&0&0&0&4 \end{bmatrix}, \begin{bmatrix} 6&0&0&0&0&0&0&0\\ 0&6&0&0&0&0&0&0\\ 0&0&6&0&0&0&0&0\\ 0&0&0&6&0&0&0&0\\ 0&0&0&0&6&0&0&0\\ 0&0&0&0&0&6&0&0\\ 0&0&0&0&0&0&6&0\\ 0&0&0&0&0&0&0&6 \end{bmatrix}.$$

Анализ устойчивости системы к отказам коммуникационных узлов при объединении серверов в кластеры

Рассмотренные варианты объединения серверов в кластеры обладают различной устойчивостью к отказам коммуникационных узлов и, в первую очередь, КНУ, осуществляющих подключение кластеров к системе.

Для базовых конфигураций без резервирования коммуникационных узлов объединение в кластеры однотипных серверов приводит к критичности системы к отказам КНУ (так как отказ любого из них вызывает нарушение связанности со всеми серверами какого-либо функционального типа) и, следовательно, к потере способности

выполнения системой каких-либо функциональных задач (запросов). При объединении в кластеры разнотипных серверов отказ любого из КНУ приводит только к нарушению связанности со всеми разнотипными серверами одного из m кластеров, остальные же кластеры суммарно обеспечивают функционирование системы при кратности резервирование каждого типа серверов, равной $m - 1$. В результате при объединении в кластеры n разнотипных по функциям серверов система гарантированно устойчива к $m - 1$ отказам не резервированных КНУ. В этом случае отказы КНУ связаны с постепенной деградацией системы (отказоустойчивое функционирование).

Таким образом, вариант комплектования кластеров разнотипными по функциональности серверами оказывается эффективней по отказоустойчивости и надежности.

При объединении в кластеры разнотипных по функциональности серверов оценим отказоустойчивость вариантов неполнофункционального комплектования кластеров, представленных матрицами $\|\varphi_{ij}\|_{n \times m}$ (2)—(7).

Для установления закономерностей рационального объединения в кластеры разнофункциональных серверов при неполной функциональности ограничимся случаями равной кратности резервирования серверов различной функциональности при ($m = n$, $r \leqslant m/2$), ($m = n$, $r > m/2$).

Для структур, представляемых матрицами вида (3), (5) при кратности резервирования серверов каждого вида $r \leqslant m/2$ и числе отказов КНУ $k < r$, система всегда сохраняет возможность функционирования. При числе отказов КНУ $k = r$ возможен отказ системы, если все отказавшие КНУ предназначены для подключения всех кластеров, соответствующих каждой отличимой комбинации построчного расположения 1 в матрице $\|\varphi_{ij}\|_{n \times m}$. Число таких комбинаций равно m/d.

Таким образом, вероятность сохранения возможности функционирования системы после $k = r$ (если m/r — целое) отказов КНУ вычисляется как

$$P(k) = \frac{C_m^r - m/d}{C_m^r}, \text{ а при } k = r + 1 \text{ как}$$

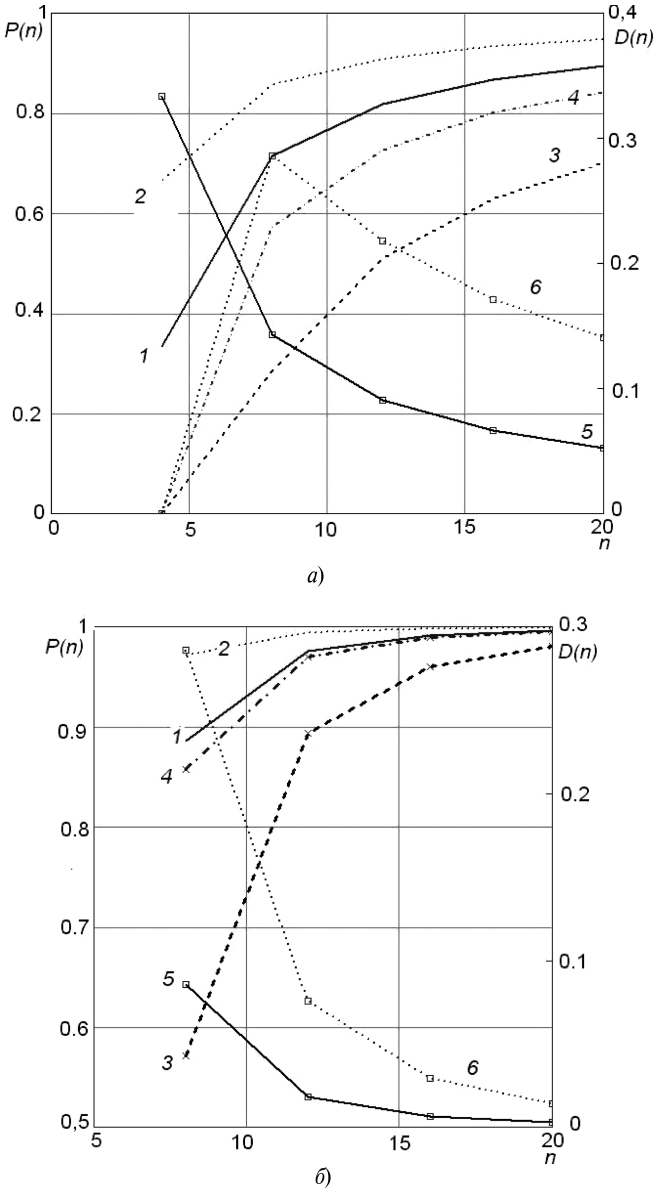
$$P(k) = \frac{C_m^k - (m/d)C_{m-r}^1}{C_m^k}.$$

При числе отказов КНУ $r \leqslant k < r + d$

$$P(k) = \frac{C_m^k - (m/d)C_{m-r}^{k-r}}{C_m^k}.$$

Анализ формул показывает, что наиболее рационален выбор смещения $d = r$, так как при этом число различных комбинаций построчного расположения 1 в матрице наименьшее и, следовательно, наименьшее число комбинаций отказов КНУ приводит к нарушению работоспособности системы.

Зависимость вероятности сохранения функционирования после возникновения k отказов КНУ от числа кластеров $m = n$ представлена на рис. 4, *а* и *б* при кратности резервирования $r = 2$ и $r = 4$ соответственно. На рис. 4, *а* и *б* кривая *1* соответствует смещению $d = 1$, а кривая *2* — смещению $d = r$ при числе отказов КНУ $k = r$. Кривая *3* соответствует смещению $d = 1$, а кривая *4* — смещению $d = r$ при числе отказов КНУ $k = r + 1$. Разницу вероятностей сохранения работоспособности поле возникновения $k = r$ отказов КНУ для



а)

б)

Рис. 4. Вероятности сохранения функционирования после отказов КНУ в зависимости от числа кластеров в системе

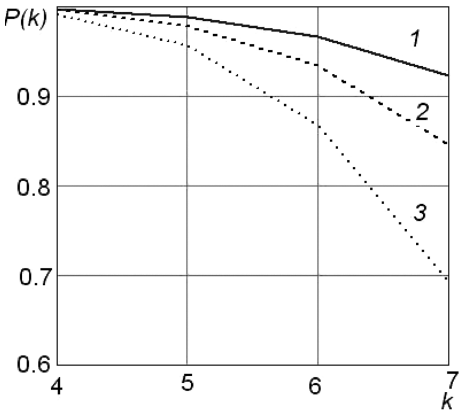


Рис. 5. Вероятности сохранения функционирования после возникновения k отказов КНУ

вариантов структуры при смещении $d = r$ и $d = 1$ представляет кривая *5*, а после $k = r + 1$ отказов — кривая *6*.

На рис. 5 представлена зависимость вероятности сохранения работоспособности системы от числа отказов КНУ при $m = n = 16$, $r = 4$, причем кривые *1*, *2* и *3* представляют случаи $d = r = 4$, $d = 2$ и $d = 1$ соответственно.

Представленные зависимости подтверждают, что для поддержки высокой отказоустойчивости системы при объединении серверов в кластеры предпочтителен выбор большего значения смещения d. Так, при одинаковых затратах на построение системы варианты объединения серверов в кластеры, представляемые матрицами вида (2)—(4), эффективней, чем представленные матрицами (5).

Таким образом, при комплектовании кластеров неполным набором функциональных типов серверов предпочтительней минимизация числа вариантов кластеров по составу функциональных типов входящих в них серверов.

Надежность компьютерной системы при объединении серверов в кластеры

При объединении в кластеры однотипных по функциональности серверов надежность системы при условии выполнения каждого функционального запроса хотя бы одним сервером (без ограничений на время ожидания запроса) вычисляется как

$$P = (1 - (1 - p_0)^{r_0}) \times \times \prod_{i=1}^{n} (1 - (1 - p_{1i})^{r_{1i}}(1 - (1 - p_{ci})^{r_{ci}}),$$

где p_0 и r_0 — вероятности работоспособного состояния и кратности резервирования КВУ; p_{1i} и r_{1i} — вероятности работоспособного состояния и кратности резервирования КНУ, используемого для подключения i-го кластера; p_{ci} и r_{ci} — вероятности работоспособного состояния и кратности резервирования серверов i-го кластера.

При идентичности по надежности всех КНУ имеем надежность

$$P = (1 - (1 - p_0)^{r_0}) \times \times \prod_{i=1}^{n} (1 - (1 - p_1)^{r_{1i}})(1 - (1 - p_{ci})^{r_{ci}}),$$

где p_1 — вероятность работоспособного состояния КНУ.

При идентичности по надежности также и всех серверов надежность системы определяется выражением

$$P = (1 - (1 - p_0)^{r_0}) \times \times [(1 - (1 - p_1)^{r_1})(1 - (1 - p_c)^{r_c})]^n,$$

где p_c и r_c — вероятности работоспособного состояния и кратность резервирования серверов в каждом кластере.

В случае комплектования каждого из m кластеров полным набором функциональных типов серверов (полнофункциональный кластер), отображаемым матрицей $\|\varphi_{ij}\|_{n \times m}$ вида (1), без резервирования КНУ и кратности резервирования КВУ, равной r_0, надежность системы оценивается как

$$P = (1 - (1 - p_0)^{r_0} \times \times \sum_{j=1}^{m} C_m^j p_1^j (1 - p_1)^{m-j} \prod_{i=1}^{n} (1 - (1 - p_{ci})^j.$$

При резервировании с кратностью r_{1i} КНУ подключения i-го кластера надежность системы

$$P = (1 - (1 - p_0)^{r_0} \sum_{j=1}^{m} C_m^j (1 - (1 - p_1)^{r_1})^j \times \times ((1 - p_1)^{r_1})^{m-j} \prod_{i=1}^{n} (1 - (1 - p_{ci})^j).$$

Если надежность всех типов серверов одинакова и равна p_c, то надежность системы

$$P = (1 - (1 - p_0)^{r_0} \sum_{j=1}^{m} C_m^j (1 - (1 - p_1)^{r_1})^j \times \times ((1 - p_1)^{r_1})^{m-j} (1 - (1 - p_{ci})^j)^n.$$

Для экспоненциального распределения времени между отказами вероятности безотказной работы соответствующих узлов в течение времени t вычисляются как $p_0 = \exp(-\lambda_0 t)$, $p_1 = \exp(-\lambda_1 t)$, $p_{ci} = \exp(-\lambda_{ci} t)$, где λ_0, λ_1, λ_{ci} — интенсивности отказов КВУ, КНУ и серверов.

Результаты расчета надежности (вероятности работоспособного состояния) компьютерной сис-

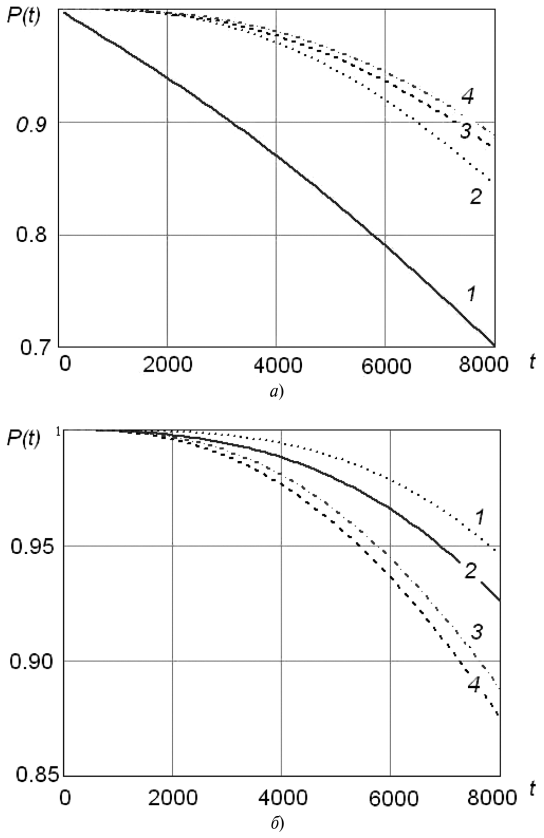


Рис 6. Надежность различных вариантов объединения серверов в кластеры

темы при различных вариантах комплектования кластеров в зависимости от времени работы системы t (ч) представлены на рис. 6, *а* и *б*. Расчеты выполнены для экспоненциального распределения времени между отказами при интенсивности отказов для коммутационных узлов (КВУ и КНУ) $\lambda_0 = \lambda_1 = 0{,}0001$ (1/ч), а для серверов всех типов $\lambda_{ci} = \lambda_c = 0{,}0005$ (1/ч).

На рис. 6, *а* кривые *1* и *3* соответствуют объединению в кластеры однотипных, а кривые *2* и *4* — разнотипных серверов. Кривые *1* и *2* соответствуют кратности резервирования КВУ, КНУ и серверов, равной $r_0 = 2$, $r_1 = 1$ и $r_c = 3$, а кривые *3* и *4* — кратности, равной $r_0 = 2$, $r_1 = 2$ и $r_c = 3$ при числе типов серверов $n = 3$ и числе кластеров (групп серверов) $m = 3$. На рис. 6, *б* кривые *1* и *3* представляют объединение в кластеры разнотипных, а кривые *2* и *4* — однотипных серверов. Кривые *1* и *2* соответствуют кратности резервирования КВУ, КНУ и серверов, равной $r_0 = 2$, $r_1 = 2$ и $r_c = 4$ при $n = m = 4$, а кривые 3 и 4 — кратности, равной $r_0 = 2$, $r_1 = 2$ и $r_c = 3$ при $n = m = 3$.

Представленные зависимости подтверждают целесообразность объединения в кластеры разнотипных по функциональности серверов при резервировании коммуникационных узлов.

Таким образом, показано, что при древовидной (двухуровневой) конфигурации коммуникационной подсистемы для обеспечения высокой надежности целесообразно подключение однотипных резервированных серверов к разным ветвям дерева (к разным коммутационным узлам нижнего уровня).

Задача оптимального резервирования узлов системы

В качестве базовых рассмотрим варианты объединения в кластеры разнотипных по функциональности серверов, которые, как показано выше, обладают лучшей отказоустойчивостью.

Задачу оптимального резервирования узлов многоуровневой компьютерной системы представим в следующей постановке.

При известной стоимости и надежности узлов требуется найти такое распределение кратности резервирования узлов системы (представляемое вектором R), которое обеспечивает минимум средств, затрачиваемых на построение системы $C(R) \to \min$ при условии обеспечения заданного нижнего уровня достигаемой надежности $P(R) \geqslant P_0$.

В качестве узлов, кратность которых определяется в результате оптимизации, рассматриваются кластеры (объединяющие разнофункциональные серверы), а также коммуникационные узлы верхнего и нижнего уровней.

При рассматриваемой древовидной структуре компьютерной системы с резервированием коммутационных узлов (см. рис. 1) стоимость ее построения определяется как

$$C(R) = c_0 r_0 + c_1 r_1 m + c_2 n m + c_3 n m r_1,$$

где c_0, c_1, c_2, c_3 — стоимости КВУ, КНУ, серверов и сетевых адаптеров (предназначенных для подключения серверов к портам КНУ); r_0, r_1, m — кратности резервирования КВУ, КНУ и кластеров; n — число функциональных типов серверов.

В результате оптимизации ищутся значения кратностей резервирования r_0, r_1, m.

Решение поставленной оптимизационной задачи может быть получено в системе математических расчетов Mathcad-14 с использованием функции Minimize.

Пусть оптимизируется система, для которой задано требование обеспечения надежности системы (вероятности безотказной работы) не меньше 0,99996. Исходные данные для поиска решения: число функциональных типов серверов $n = 4$, вероятности безотказной работы коммутационных узлов 0,96, серверов 0,86, а также стоимости компонент $c_0 = 1$ (у. е.), $c_1 = 1$ (у. е.), $c_2 = 5$ (у. е.), $c_3 = 0{,}3$ (у. е.). В этом случае мини-

мум стоимости системы (C = 145 у. е.) достигается при искомых значениях кратности резервирования $r_0 = 4$, $r_1 = 2$ и числе полнофункциональных кластеров $m = 6$.

Заключение

Таким образом, поставлена и решена задача оптимизации объединения в кластеры резервированных серверов различного функционального назначения. Показано, что при древовидной (двухуровневой) конфигурации коммуникационной подсистемы с резервированием для обеспечения высокой отказоустойчивости и надежности системы целесообразно объединение в кластеры разнотипных по функциональности серверов.

Установлено, что при комплектовании кластеров неполным набором функциональных типов серверов предпочтительней минимизация числа вариантов кластеров по составу функциональных типов входящих в них серверов.

Список литературы

1. **Ретана Ф.** Принципы проектирования корпоративных IP-сетей. М.: Вильямс, 2002.
2. **Проектирование** кампусных сетей http://crystal.crystalnet.ru.
3. **Богатырев В. А.** Надежность многоуровневой дублированной отказоустойчивой коммуникационной подсистемы // Вестник компьютерных и информационных технологий. 2008. № 4.
4. **Богатырев В. А.** Надежность и эффективность резервированных компьютерных сетей. // Информационные технологии. 2006. № 9.

УДК 004.738

Р. Э. Асратян, канд. техн. наук,
ведущий науч. сотр.,
Институт проблем управления
им. В. А. Трапезникова РАН, г. Москва,
e-mail: rea@19.ipu.rssi.ru

Метод организации Web-сервисных взаимодействий между удаленными частными сетями

Рассматривается проблема применения технологии Web-сервисов для организации взаимодействия между узлами, расположенными в разных частных сетях, объединенных глобальной сетью. Описывается подход к решению этой проблемы, основанный на организации защищенного туннеля с использованием специальной сетевой службы, обеспечивающей возможность межсерверного взаимодействия и межсерверной маршрутизации данных в сети, защиту от несанкционированного доступа и возможность совмещения передачи данных через туннель с их обработкой.

Ключевые слова: *распределенные системы, Интернет-технологии, Web-сервисы.*

Введение

Технология Web-сервисов привлекает все большее внимание разработчиков распределенных систем в качестве средства организации сетевых взаимодействий [1]. Языковая и платформенная независимость этой технологии, поддержка в ряде современных средств разработки (MS Visual Studio, Borland C++ Builder и др.) позволили ей стать одним из основных инструментов разработки систем и приложений в архитектуре SOA (Service-oriented architecture).

Тем не менее, технология Web-сервисов не свободна от недостатков, которые особенно ощутимо проявляются в разработках больших распределенных систем (т. е. систем, включающих десятки и сотни взаимодействующих компонентов, удаленных друг от друга на сотни и тысячи километров):

- Web-сервисы не содержат встроенных средств защиты от несанкционированного доступа;
- технология Web-сервисов не поддерживает собственных средств межсерверного взаимодействия и межсерверной маршрутизации данных в сети. Это означает, что клиент может требовать обслуживания только у того сервера, с которым он непосредственно соединился, а взаимодействие через серверы-посредники не предусматривается. Данное ограничение может оказаться чрезмерно жестким для распределенных систем, предназначенных для функционирования в неоднородных сетях (т. е. в сетях, не обеспечивающих возможность прямого сетевого соединения между любыми двумя узлами).

Рассмотрим корпоративную сеть, представляющую собой множество удаленных друг от друга и независимо администрируемых частных локальных сетей, объединенных глобальной сетью через "пограничные" (т. е. присоединенные сразу к двум сетям) узлы-маршрутизаторы (именно эту ситуацию мы и будем иметь в виду в данной статье). Предположим, что по соображениям безопасности и/или вследствие нехватки реальных IP-адресов рабочие сетевые узлы (и клиентские, и серверные) "спрятаны" в этих частных сетях.

Обычный способ организации взаимодействия между узлами из разных частных сетей через глобальную сеть заключается в построении защищенных VPN-туннелей между частными сетями [2]. Однако данный подход требует уникальности частных IP-адресов серверов в пределах всей корпоративной сети (очевидно, что если два или более серверов в разных частных сетях имеют один и тот же IP-адрес, то VPN-туннель сможет обеспечить доступ лишь к какому-то одному из них). Если число и размеры частных сетей велики, а администрируются они действительно независимо, это ограничение может оказаться чересчур жестким.

В статье рассматривается альтернативный подход к решению данной задачи. Этот подход также основан на построении "туннелей" через глобальную сеть, но в данном случае они строятся не на уровне IP-маршрутизации, а на уровне специальной Интернет-службы, основанной на TCP/IP. Следует сразу оговориться, что если VPN-туннель способен поддерживать любые IP-взаимодействия, то описываемый подход ориентирован только на поддержку взаимодействий по протоколу HTTP (на котором основана вся Web-технология, включая и технологию Web-сервисов).

В работе [3] описана сетевая служба RECS (*Remote Executable Call Service* — служба удаленного вызова исполняемых модулей), предназначенная для поддержки распределенных вычислений. Важной особенностью этой службы является наличие встроенных средств межсерверного взаимодействия и межсерверной маршрутизации данных, а также средств защиты от несанкционированного доступа. В данной статье речь идет о частном применении службы RECS для построения сетевых туннелей, поддерживающих HTTP-взаимодействия. Идея подхода основана на использовании HTTP-взаимодействий в пределах частных локальных сетей и переходе на RECS-взаимодействия для обменов данными в глобальной сети (рис. 1). Разумеется, главное требование к туннелю заключается в его "прозрачности" и для HTTP-клиентов, и для HTTP-серверов.

Главными преимуществами RECS-туннелей по сравнению с VPN-туннелями являются:

- допустимость повторения IP-адресов серверов в связываемых частных сетях;
- возможность простого совмещения передачи данных через туннель с их обработкой (сжатием, шифрованием и т. п.);
- более простое и безопасное создание тоннеля (создание VPN-туннеля средствами службы IP-маршрутизатора [2] является достаточно сложной операцией, а некорректное администрирование на данном уровне может полностью разрушить работу сети).

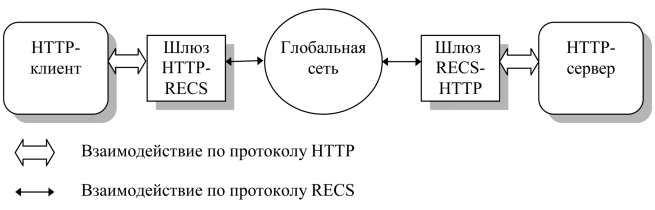


Рис. 1. RECS-туннель

Краткие сведения о RECS

Все современные развитые командные процессоры позволяют осуществить одновременный запуск цепочки исполняемых модулей для решения одной задачи, связав стандартный вывод каждого модуля со стандартным вводом следующего в цепочке (этот механизм берет начало от первых версий системы UNIX). Интернет-служба RECS является "сетевым обобщением" этого механизма. Она позволяет пользователю осуществить одновременный вызов сразу нескольких исполняемых модулей (обработчиков), расположенных на разных сетевых узлах. На поведение каждого обработчика накладывается единственное ограничение: он должен читать входные данные со стандартного ввода и писать выходные в стандартный вывод (возможна также передача ему параметров командной строки). В момент вызова непосредственно обслуживающий клиента RECS-сервер (далее будем называть его "локальным") организует запуск указанных обработчиков на собственном узле и/или на удаленных узлах (разумеется, также оснащенных RECS-серверами). В тот же момент усилиями всех задействованных серверов создается множество "каналов", попарно соединяющих запущенные обработчики по принципу "стандартный вывод предыдущего со стандартным вводом следующего". Конечно же, стандартный ввод первого обработчика и стандартный вывод последнего соединяются с клиентом. С этого момента все отправленные клиентом данные будут проходить последовательную обработку во всех запущенных исполняемых модулях, а результирующие данные будут немедленно направляться клиенту по мере их появления. На содержание данных (текстовых или двоичных) и последовательность их отправки и приема не накладывается никаких ограничений (хотя, разумеется, поведение клиентов должно быть логически согласовано с поведением обработчиков).

Понятие обобщенного "канала взаимодействия", соединяющего обработчики друг с другом (а также крайние обработчики с клиентом), является основополагающим для RECS. Хотя для его организации в разных ситуациях используются разные технологии — "трубопровод" (*pipe*), если

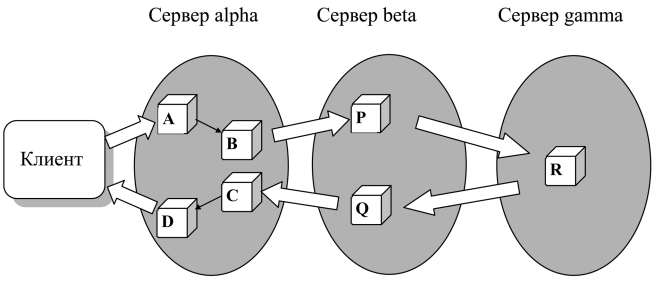


Рис. 2. Обработка вызова Process("A, B, beta$(P, gamma$R, Q), C, D")

соединяемые обработчики находятся на одной машине, и межсерверное сетевое взаимодействие, если на разных, — у пользователя создается иллюзия единого связующего механизма, достаточно ясного и простого в обращении.

Как и всякая служба, основанная на TCP/IP [3, 4], RECS поддержана клиентским и серверным программным обеспечением (ПО). Сервер RECS представляет собой постоянно активную многопоточную программу, обслуживающую запросы на обработку от клиентов или других серверов. Клиентское ПО представляет собой библиотеку функций (методов специального класса RECSClient), реализующих прикладной программный интерфейс (API) к RECS.

На рис. 2 отображен чисто иллюстративный пример распределенной обработки с помощью RECS. Предполагается, что клиентская программа установила соединение с сервером alpha и выполнила вызов функции.

Process ("A, B, beta$(P, gamma$R, Q), C, D");

В качестве аргумента функции передается командная строка, содержащая последовательность имен обработчиков и серверов (символ "$" отделяет имена серверов от имен обработчиков). Обрабатывая командную строку, сервер alpha выполнит запуск обработчиков A и B (отображены "кубиками"), после чего обнаружит "вложенную" командную строку **beta$(P, gamma$R, Q),** которая должна выполняться на сервере beta. Поэтому сервер alpha осуществит обращение к серверу beta. Аналогично сервер beta, в свою очередь, обнаружит "вложенную" командную строку **gamma$R** и осуществит обращение к серверу gamma для запуска единственного обработчика R. В итоге обработчики A, B, C и D будут запущены на сервере alpha, обработчики P и Q — на сервере beta, а обработчик R — на сервере gamma.

Важно подчеркнуть, что запуск обработчиков всегда сопровождается созданием каналов взаимодействия или с помощью "трубопровода" (простые стрелки), или на основе TCP-соединения (фигурные стрелки). Предположим также, что обработчик A последовательно считывает символы со своего стандартного ввода и немедленно отправляет их в стандартный вывод, перекодируя при этом символ "a" из нижнего регистра в верхний. Аналогично обработчик B делает то же самое, но перекодирует символ "b", обработчик C — символ "c" и т. д.

Если теперь клиентская программа осуществит отправку строки **a query must be processed** на сервер alpha (с помощью специальной функции SendString), а потом прочтет результат обработки (также с помощью специальной функции GetString), то она получит следующую строку результата: **A Query must Be PRoCesseD.** Важно подчеркнуть, что обработка строки в этом случае проводится семью обработчиками на трех серверах.

Как видно из приведенного примера, RECS-сервер имеет дело с объектами трех типов:

- *клиенты* — источники запросов на обслуживание;
- *обработчики* — программы (исполняемые модули), запускаемые сервером для обработки данных;
- *серверы* — другие RECS-серверы, доступные для взаимодействия.

Объекты всех типов должны быть предварительно зарегистрированы на сервере RECS. Другими словами, сервер отвергает запросы, исходящие от незарегистрированных клиентов или агентов, или же запросы на запуск незарегистрированных обработчиков. При регистрации объекту присваивается уникальное имя; кроме того, с ним связывается определенная управляющая информация (пароль, права доступа, спецификация исполняемого модуля обработчика и т. п.). Отметим, что обращение клиента к серверу сопровождается "двусторонней проверкой подлинности" (защита сервера от неавторизованного клиента и защита клиента от фальсифицированного сервера) и "скрытой" передачей пароля по сети на основе примерно той же идеи, на которой построен протокол MS-Chap v2 [2]. Точно такая же проверка осуществляется при обращении сервера к серверу (в этом случае обращающийся сервер выступает в роли клиента). Регистрационная запись удаленного сервера содержит его IP-адрес (или Интернет-имя), сетевой порт, а также имя и пароль пользователя, от имени которого будут проводиться обращения к удаленному серверу при межсерверных взаимодействиях.

Принципы организации RECS-туннеля

Рассмотрим множество частных локальных сетей (LAN), объединенных глобальной сетью, и предположим, что проектируемая распределенная система требует организации взаимодействия между узлами, размещенными не только в одной ча-

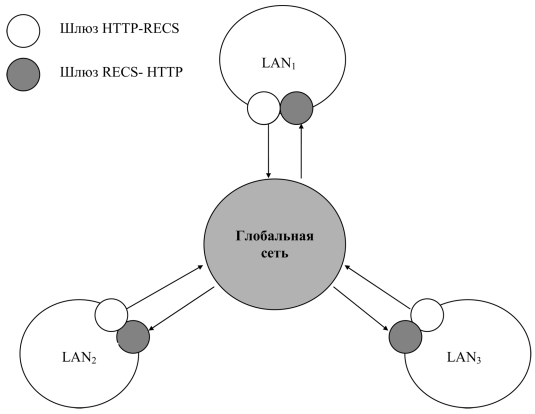


Рис. 3. Шлюзы между HTTP и RECS

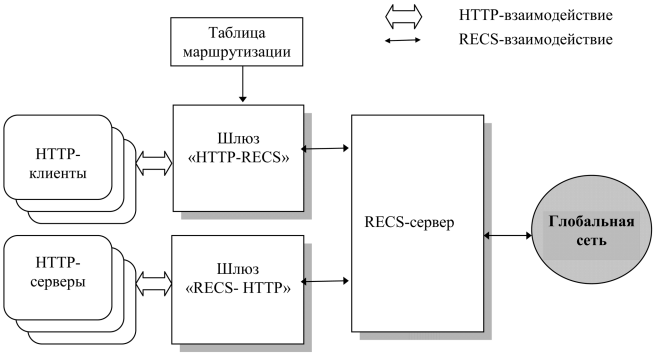


Рис. 4. Шлюзы HTTP-RECS, RECS-HTTP и RECS-сервер

стной сети, но и в разных. Объединение частных сетей через глобальную сеть предполагает наличие в каждой из них по крайней мере одного "пограничного" узла, соединенного и с локальной, и с глобальной сетями и выполняющего функцию IP-маршрутизатора [4]. Будем считать, что именно на таких узлах устанавливаются программные средства шлюзов HTTP-RECS и RECS-HTTP, обеспечивающих передачу и прием потоков HTTP-запросов и HTTP-ответов с помощью RECS (рис. 3). Кроме того, на каждом из таких узлов должен быть установлен RECS-сервер, находящийся в "коллективном" пользовании шлюзов обоего типа (рис. 4). Несмотря на кажущуюся "симметрию" способы реализации шлюзов различны: шлюз HTTP-RECS реализуется как специализированный прокси-сервер, обеспечивающий "перехват" исходящих соединений от всех HTTP-клиентов обслуживаемой локальной сети, а шлюз RECS-HTTP — как обработчик, запускаемый RECS-сервером по запросам от клиентов или других серверов.

Шлюз HTTP-RECS представляет собой постоянно активную многопотоковую программу, выполняющую постоянное слежение за входящими соединениями на выделенном ей сетевом порте [4]. Главная задача шлюза — обеспечить перевод HTTP-взаимодействий "на язык" RECS. Шлюз приводится в действие путем глобальной настройки всех HTTP-клиентов в локальной сети на работу через "прокси-сервер" с указанием IP-адреса и порта шлюза в качестве параметров настройки. В результате все исходящие соединения от HTTP-клиентов (независимо от URL адресуемого Интернет-ресурса) будут поступать непосредственно "на вход" шлюза. После обнаружения каждого соединения шлюз порождает отдельный поток (*thread*), в рамках которого выполняется вся последующая обработка.

Обработка каждого входящего HTTP-соединения в шлюзе HTTP-RECS опирается на таблицу маршрутизации шлюза, которая связывает доменные имена адресуемых Интернет-ресурсов с именами удаленных RECS-серверов и IP-адресами узлов в удаленных локальных сетях. С помощью таблицы маршрутизации шлюза выполняется первый шаг маршрутизации, последующие шаги реализуются средствами RECS.

Обработка входящего соединения включает следующие основные шаги:

- прием HTTP-запроса от HTTP-клиента по сети;
- извлечение Интернет-имени удаленного узла из URL адресуемого Интернет-ресурса (Web-страницы, Web-сервера и т. п.) и поиск записи, соответствующей этому имени в таблице маршрутизации шлюза. Если такая запись отсутствует, шлюз устанавливает соединение с удаленным узлом без "перехода" на протокол RECS и выполняет обработку данного входящего соединения в режиме классического прокси-сервера (данный режим мы рассматривать не будем);
- извлечение командной строки RECS и IP-адреса адресуемого ресурса в удаленной локальной сети из найденной строки таблицы маршрутизации. Командная строка RECS должна содержать полное имя шлюза RECS-HTTP на удаленном сервере (т. е. имя RECS-сервера и регистрационное имя шлюза в форме "имя_сервера$имя_шлюза"). Полученный из таблицы IP-адрес ресурса помещается в заголовок HTTP-запроса;
- соединение с RECS-сервером и передача ему полученной командной строки в качестве параметра функции Process. Отправка HTTP-запроса RECS-серверу (для передачи его удаленному шлюзу RECS-HTTP по глобальной сети);
- ожидание окончания обработки и прием HTTP-ответа от RECS-сервера;
- отправка HTTP-ответа клиенту по входящему соединению.

Во все время обработки входящего соединения HTTP-клиент находится в состоянии ожидания.

Таблица маршрутизации шлюза представляет собой текстовый файл, каждая строка которого имеет следующий формат:

"Имя ресурса" "Командная строка RECS-сервера" "IP-адрес ресурса"

Имя ресурса играет роль ключа при поиске строки, а остальные реквизиты представляют собой результат поиска. Рассмотрим следующий пример таблицы:

alpha Srvl$RecsHttp 192.168.0.10
beta Srv2$ RecsHttp 192.168.0.10
gamma Srv3$ RecsHttp 192.168.0.20

При обработке запроса к ресурсу **http://beta/document1** будет найдена вторая строка таблицы. В результате HTTP-запрос будет передан по глобальной сети в удаленный RECS-сервер Srv2, где к нему будет применен обработчик с регистрационным именем RecsHttp — шлюз RECS-HTTP. Этот шлюз должен обеспечить обращение к ресурсу **http://192.169.0.10/document1** в обслуживаемой им локальной сети по протоколу HTTP.

Шлюз RECS-HTTP выполняет обратную задачу по отношению к шлюзу HTTP-RECS. По своему "статусу" он представляет собой специализированный обработчик в RECS-сервере. Это означает, что он получает управление в результате выполнения в RECS-сервере командной строки, в которой обозначено его имя. Обработка HTTP-запроса в шлюзе включает следующие шаги:

- считывание HTTP-запроса из стандартного ввода;
- просмотр заголовка HTTP-запроса и извлечение IP-адреса адресуемого ресурса в локальной сети;
- установление соединения с Web-сервером на соответствующем узле по локальной сети;
- передача HTTP-запроса по установленному соединению;
- ожидание окончания обработки запроса, прием HTTP-ответа и запись его в стандартный вывод.

Доставка HTTP-ответа по глобальной сети в исходный RECS-сервер осуществляется средствами RECS.

В заключение коротко остановимся на возможности совмещения передачи данных через туннель с их обработкой. Такая возможность может быть реализована путем включения дополнительных обработчиков в командные строки. Поясним это на простом примере. Предположим, что в нашем распоряжении имеются два обработчика с регистрационными именами Zip и Unzip, причем первый из них реализует потоковую компрессию данных, считанных со стандартного ввода, и отправку их в стандартный вывод, а второй — декомпрессию. Если заменить командную строку "SrvRecsHttp" в таблице маршрутизации шлюза HTTP-RECS на строку "Zip, Srv$(Unzip, RecsHttp, Zip), Unzip", то будут выполняться автоматическая компрессия HTTP-запросов и HTTP-ответов перед отправкой их в глобальную сеть и декомпрессия — после получения из глобальной сети без ущерба для "прозрачности" туннеля. (Подчеркнем, что данный прием дает возможность сократить сетевой трафик даже в том случае, когда Web-сервер и/или Web-браузер не содержат встроенных средств компрессии/декомпрессии данных.)

Заключение

Безусловно, взаимодействие через RECS-туннель (как и всякое взаимодействие через посредников) связано с дополнительными временны́ми задержками по сравнению с прямым взаимодействием клиентов и серверов. Эти задержки могут быть сведены к минимуму путем реализации шлюза RECS-HTTP в составе самого RECS-сервера, а не в форме внешнего исполняемого модуля (обработчика). Данный подход несколько противоречит принципам организации RECS, но позволяет получить ощутимый практический эффект.

В настоящее время программные средства поддержки RECS-тоннеля реализованы в операционной среде Win32. Проведенные эксперименты подтвердили применимость и эффективность подхода для обслуживания разных форм HTTP-взаимодействий:

- "тонких" HTTP-клиентов (Web-браузеров) с Web-сайтами;
- "толстых" HTTP-клиентов (WS-клиентов) с Web-сервисами.

Важно отметить, что встраивание RECS-туннелей в целый ряд уже разработанных распределенных систем и приложений ни в одном случае не потребовало какой-либо адаптации последних, что подтверждает выполнение требования "прозрачности" туннеля для взаимодействующих сторон.

Список литературы

1. **Шапошников И. В.** Web-сервисы Microsoft. NET. СПб.: БХВ-Петербург, 2002. 336 с.

2. **Андреев А. Г., Беззубов Е. Ю., Емельянов М. М. и др.** Windows 2000: Server и Professional. СПб.: "БХВ-Санкт-Петербург", 2001. 1055 с.

3. **Асратян Р. Э.** Интернет-служба для поддержки распределенных вычислений // Информационные технологии. 2006. № 12. С. 60—66.

4. **Джамса К., Коуп К.** Программирование для Интернет в среде Windows. СПб.: "Питер", 1996. 659 с.

УДК 001.893:005

А. Ю. Силина, аспирант,
В. Д. Васильева, канд. техн. наук, доц.,
В. Е. Дербишер, д-р хим. наук, проф.,
Волгоградский государственный технический университет
И. В. Гермашев, канд. техн. наук, доц.,
Волгоградский государственный педагогический университет

Систематизация наукометрических показателей эффективности научной деятельности

Проведена систематизация наукометрических показателей, применяемых для оценки научной деятельности отдельных специалистов, научных сообществ, областей научной деятельности и вычисления рейтинга научной печатной продукции. Дана развернутая характеристика каждого показателя и область его применения.

Ключевые слова: *наукометрические показатели, оценка научной деятельности, рейтинг научной печатной продукции.*

Введение

В условиях активно развивающейся мировой научно-технической деятельности большую роль для изучения ее различных аспектов в последнее время стала играть сравнительно недавно появившаяся отрасль знаний — наукометрия, развивающая методы количественного и качественного анализа научной деятельности в самом широком смысле. Необходимость такого изучения не вызывает сомнений, поскольку позволяет заинтересованным сторонам составить определенную картину о количественном и качественном наполнении различных направлений научной деятельности в рамках отдельного ученого, научной организации, региона, страны и т. д. и, владея данными знаниями и руководствуясь определенной целью, проводить эффективную научную политику и управлять научными направлениями вплоть до отдельных исследований.

С методами количественного анализа научной деятельности, прежде всего, связывают наукометрические показатели, в последнее время находящие применение, в том числе, благодаря созданию международных и национальных баз данных [1—4], концентрированно владеющих и накапливающих обширную предметную информацию. Однако применяемые наукометрические показатели относительно субъективны, разрознены, и мы не обнаружили в литературе их систематизации. В то же время их анализ и систематизация явились бы важным средством повышения теоретического и практического значения этих показателей, по крайней мере, тех из них, которые являются наиболее содержательными. Решению данной задачи, а именно анализу существа наукометрических показателей и их систематизации, и посвящена настоящая работа.

Классификация наукометрических показателей

Прежде всего установим объекты научной деятельности, связанной с созданием новой научной информации, которые можно подвергнуть наукометрическому исследованию. Объекты научной деятельности можно сгруппировать по следующим признакам: принадлежности научному сообществу, областям научной деятельности, видам научной печатной продукции. Без детализации это показано на рис. 1.

Представленные на рис. 1 группы объектов научной деятельности отражают многообразие применяемых и потенциальное разнообразие еще не разработанных наукометрических показателей, полный комплекс которых в полной мере позволит проводить эффективный анализ всех сфер научной деятельности.

Применение этих показателей для оценки деятельности *научного сообщества* позволяет позиционировать ученых, исследовательские центры (например, лаборатории, кафедры, временные творческие коллективы и др.), научные организации (академический вуз, НИИ) в локальной и мировой научных системах. Сравнительный анализ дает возможность оценивать вклад исследователей, как производителей новой информации,

Рис. 1. Классификация научной деятельности как объекта наукометрического анализа

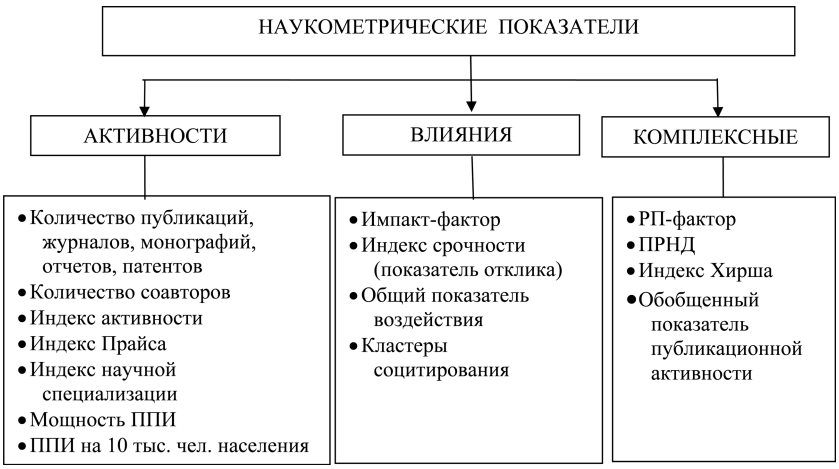


Рис. 2. Классификация наукометрических показателей

в мировой информационный массив, изучать взаимосвязи между отдельными сообществами, проводить мониторинг научной деятельности во времени.

Можно отметить также, что анализ продуктивности различных *областей знания*, выраженной количеством научной информации, уже сейчас в определенной мере позволяет выявлять быстро развивающиеся области, получать некоторые представления о внутренней (логической) структуре фронта научных исследований, выявлять зарождающиеся и перспективные направления, прослеживать их динамику и оценивать эффективность исследовательских программ, а также изучать проникновение новых методов исследования в смежные области знаний.

Из общего контекста научно-технической деятельности и государственной политики следует, что исследование и оценка *научных изданий*, являющихся результатом научной деятельности и ответственных за хранение и распространение научной информации, так же, как и анализ функционирования социума в целом, позволяет изучать процессы движения информации во времени и пространстве внутри научного сообщества (социума), совершенствовать научные коммуникации, оптимально комплектовать журнальный фонд научных библиотек, способствовать развитию этой деятельности и управлять ею.

Проведенный анализ применяемых в настоящее время наукометрических показателей позволил сгруппировать их следующим образом: количественные показатели *активности* научной деятельности, качественные показатели *влияния* (*цитируемости*) публикаций на информационный массив и комплексные показатели, учитывающие количественные и качественные критерии оценки эффективности научной деятельности. Предлагаемая нами классификация представлена на рис. 2.

Показатели *активности* являются динамическими, и в первую очередь, отражают интенсивность, с которой ученые (организации) публикуют статьи либо другие научные материалы, включая документы на интеллектуальную собственность (заявка на изобретение, патент, официальная регистрация программы для ЭВМ и др.).

Показатели *влияния* отражают качественные аспекты научной деятельности, оценивают степень полезности (использования) научных идей для других ученых и специалистов в генерировании ими новых результатов исследований и разработок. Наиболее распространенной оценкой показателя влияния научной статьи сегодня считается цитирование, т. е. число ссылок на данную работу других авторов, что отражает ее значение. Чем больше статья цитировалась, тем большее влияние она оказала на научное сообщество, тем более эффективной может считаться научная деятельность ученого.

Комплексные наукометрические показатели включают как количественную составляющую (показатели активности), так и качественную (показатели влияния) и предназначены для более объективной и полной оценки научной деятельности.

Подробная характеристика используемых в настоящее время наукометрических показателей представлена в таблице.

Таблица

Характеристика наукометрических показателей

Наукометрические показатели	Расчетная формула	Назначение	Год издания
Показатели активности			
Индекс активности	$IA = D_r/D_s$, где D_r — доля определенной дисциплины в общем массиве публикаций региона; D_s — доля этой же дисциплины в общем массиве публикаций всей страны	Определение публикационной активности региона по какой-либо дисциплине относительно публикационной активности страны [5]	2002
Индекс Прайса	$IP = m/(n - m)$, где m — число ссылок на оперативную литературу (возрастом менее 5 лет); $(n - m)$ — число ссылок на архивную литературу (возрастом более 5 лет)	Оценка журнала, института и определенного индивида или отдельной страны, определяющая фронт научных разработок [6]	1971

Продолжение таблицы

Наукометрические показатели	Расчетная формула	Назначение	Год издания
Индекс научной специализации	$ISS_{it} = \frac{A_{it}}{\Sigma_i A_{it}} / \frac{WA_{it}}{\Sigma_i WA_{it}}$, где ISS_{it} — индекс научной специализации страны в области i в году t; A_{it} — число статей в i, принадлежащих национальным авторам, в научных журналах, реферируемых в базе данных *SCI/SSCI*, в году t; WA_{it} — общее число статей в области i в научных журналах, реферируемых в базе данных *SCI/SSCI* в году t	Характеристика "научных интересов" стран путем сравнения структуры национальных и мировых публикаций [7]	2007
Мощность потока патентной информации	$M_{\text{ППИ}} = \Sigma П_5$, где $П_5$ — число патентов по странам проживания патентообладателей за последние 5 лет	Оценка мировой картины распределения патентной активности в предметной области [8]	1960—1970
Поток патентной информации на 10 тыс. чел. населения	$N_{\text{ППИ}} = П_5/5 \cdot 10\,000$, где $П_5/5$ — среднегодовое число выданных патентов за последние 5 лет, приходящихся на 10 тыс. чел. населения	Оценка массовости изобретательского творчества населения рассматриваемой страны [8]	1960—1970
Индекс активности патентирования	$A_{\text{ППИ}} = \Sigma П_5/\Sigma П_{10} \cdot 100\ \%$, где $\Sigma П_5$ — число патентов за 5 лет; $\Sigma П_{10}$ — число патентов за 10 лет	Выявление наиболее активных стран, работающих в исследуемой области в последние года [8]	1960—1970
Показатели влияния			
Импакт-фактор (индекс цитирования)	$IF = q/M$, где q — общее число ссылок в общем потоке на данный журнал за предшествующие 2 года; M — число опубликованных статей в данном журнале за эти же 2 года	Определение уровня конкурентоспособности (рейтинг) журнала, ученого, различных научно-образовательных организаций [9]	1963
Индекс срочности (показатель отклика, Immideacy Index)	$Im.\ Ind = CIT(Y, Y)/PUB(Y)$, где $CIT(Y, Y)$ — число ссылок на статьи журнала, опубликованные в прошлом году; $PUB(Y)$ — общее число статей	Определение скорости опубликования научных работ [10]	1975
Общий показатель воздействия (метод Маршаковой—Шайкевич)	$Ig = SR/SS$, где SR — число ссылок, полученных журналами в текущем году на статьи, опубликованные в них за предыдущие 2 года; SS — число статей в этих журналах за тот же период	Оценка вклада различных стран в мировую науку [11]	1995
Кластеры социтирования (метод Маршаковой—Смолла)	Совокупность высокоцитируемых публикаций, которые цитировались совместно в третьих работах	Определение кластеров ("фронтов") научной активности, составление "карты" изучаемой дисциплины и выявление нарождающихся областей исследования, оценка научной деятельности [12, 13]	1973
Комплексные показатели			
РП-фактор	$РП = 1000 \cdot \sum_{i-1}^{N} \frac{Im_i}{S_i + 1}$, где Im_i — импакт-фактор i-го журнала; N — число статей; $Si + 1$ — полное число авторов статьи или монографии	Оценка уровня ученого, учитывающая число опубликованных статей, импакт-фактор журнала и число соавторов [14]	2007
ПРНД (показатель российской научной деятельности)	$ПРНД = kG + pM + rU + hD + sK + bP + gR$, где G — публикации в журналах; M — монографии; U — учебники; D — доклады на конференциях; K — научно-образовательные курсы; P — патенты; R — научное руководство; k, p, r, h, s, b, g — весовые коэффициенты	Определение результативности научной деятельности ученого [15]	2006
Индекс Хирша	h-индекс отражает число публикаций, каждая из которых процитирована не менее h раз	Оценка продуктивности ученого, основанная на числе публикаций одного автора и числе цитирований этих публикаций [16]	2005
Обобщенный показатель публикационной активности (Силина, Васильева, Гермашев, Дербишер)	$\begin{cases} S = \sum_{i=1}^{n-1} S_i, \\ U = \frac{1}{2} r_{\text{м}}^2 \varphi_{\text{м}}, \end{cases}$ где S — суммарное число публикаций; U — относительный показатель качества опубликованного научного материала	Определение квалификационного потенциала ученого и его общественного признания. Показатель учитывающий публикации, а также учебники, учебные пособия и монографии, патенты и лицензии	2007

Из приведенных в таблице показателей наиболее популярными являются *импакт-фактор* и *индекс Хирша*, основанные на анализе цитирования публикаций. На базе импакт-фактора предложен целый ряд схожих показателей для разного рода оценок с привлечением больших массивов данных, методов математической статистики и моделирования —это *кластеры социтирования*, *общий показатель воздействия, индекс срочности* и др. Можно отметить также, что в зарубежных исследованиях часто применяется *суммарный индекс цитирования*, в котором суммируются ссылки, сделанные на работы конкретного исследователя как другими авторами, так и самим исследователем (так называемое самоцитирование). В некоторых случаях (например, конкурс International Soros Science Education Program) учитывается лишь индекс цитирования за минусом самоцитирования и даже цитирования теми, кто в разное время был соавтором этого исследователя. *Индекс Хирша* является комплексной характеристикой продуктивности ученого, претендующий на более высокую точность и объективность по сравнению с импакт-фактором. Данный показатель нашел широкое применение в области электронных документов. *Индекс Прайса* учитывает две популяции ссылок, частично перекрывающие друг друга. С одной стороны, это ссылки на "архивную" литературу (свыше 15—20-летней давности), распределенные довольно равномерно и медленно уменьшающиеся по мере старения литературы. С другой стороны, это ссылки на литературу "оперативного воздействия", сравнительно современную и находящуюся на переднем крае исследований.

Достаточно широко используются показатели количества научных работ, что свидетельствует об актуальности какой-либо темы. Изучение частоты публикаций позволяет выявить "очаги научной активности", оценить распределение ресурсов, выявить лидеров (страны, институты и т. п.). В данную категорию показателей можно включить *индекс активности*, используемый при оценке вклада (в научном плане) отдельной страны в мировой науке, и индекс научной специализации различных стран.

Особую группу представляют комплексные наукометрические показатели, разработанные отечественными исследователями. К ним относятся *РП-фактор*, *ПРНД* и *обобщенный показатель публикационной активности*, последний из которых является разработкой коллектива авторов данной статьи. Широкое применение приведенных показателей пока ограничивается отечественными исследованиями в данной предметной области.

Заключение

Проведенная систематизация наукометрических показателей, применяемых для оценки эффективности научно-технической деятельности, показывает, что здесь пока объективно не выработано единых подходов, но имеется хорошая основа для развития данного направления, в частности для создания ограниченного набора комплексных характеристик.

Кроме того, представленные материалы несколько упрощают выбор показателей, отвечающих конкретным задачам наукометрического исследования в предметных областях. Такое активное использование наукометрии при управлении и регулировании научной деятельностью позволит повысить ее предметность и эффективность за счет, например, рационального стимулирования в виде планового финансирования, выделения грантов, организации новых научных направлений и коллективов и т. д. Учитывая же, что наукометрия не только дает статистический материал, отражающий ретроспективу, но и обладает прогностическими возможностями, развитие ее становится бесспорно очень важным в современной обстановке достаточно плотной научно-технической конкуренции.

Список литературы

1. **Rousseau R. L.** Journal Evaluation: Technical and Practical Issues // Library Trends. 2002. V. 50. I. 3. P. 418—439.

2. **Jin B., Zhang J., Chen P., Zhu X.** Development of the "Chinese Scientometric Indicators" (CSI) // Scientometrics. 2002. V. 54. I. 1. — P. 145—154.

3. **Negishi M., Sun Y., Shigi K.** Citation Database for Japanese Papers: A new bibliometric tool for Japanese academic Society // Scientometrics. 2004. V. 60. I. 3. — P. 333—351.

4. **Российский** индекс научного цитирования [Электронный ресурс]. — [2008]. — Режим доступа: http://www.elibrary.ru.

5. **Bordons M., Fernandez M. T., Gomez I.** Advantages and limitations in the use of impact factor measures for the assessment of research performance in a peripheral country // Scientometrics. 2002. V. 53. I. 2. — P. 195—206.

6. **Прайс Д.** Квоты цитирования в точных и неточных науках, технике и ненауке // Вопросы философии. 1971. № 3. — С. 149—155.

7. **Российская** наука: библиометрические индикаторы [Электронный ресурс]. — [2007]. — Режим доступа: http://ecsocman.edu.ru

8. **Kind D. V.** Patent serve as a unique information source // WPI. — 1983. — N 3. — P. 135—136.

9. **Garfield E., Sher I. H.** New Factor in the Evaluation of Scientific Literature Through Citation Indexing // American Documentation. — 1963. — V. 14. I. 3. — P. 195—201.

10. **Computer** Horizons. Subject classification and influence weights for 2300 journals. // Contract. — 1975. — N 6. — P. 627—629.

11. **Маршакова-Шайкевич И. В.** Вклад России в развитие науки: библиометрический анализ. — М.: Янус, 1995. — 248 с.

12. **Маршакова И. В.** Система связей между документами, построенная на основе ссылок // НТИ. Сер. 2. — 1973. — № 6. — С. 3—8.

13. **Small H. G.** Co-citation in the scientific literature a new measure of the relationship between two documents. // J. Amer. Soc. Inform. Sci. — 1973. — V. 24. N 4. — P. 265—269.

14. **Русский** переплет [Электронный ресурс]. — [2007]. — Режим доступа: http://www.pereplet.ru/nauka/young/

15. **ПРНД** или Особенности оценки национальной науки [Электронный ресурс]. — [2008]. — Режим доступа: http://www.elibrary.ru/proiects/egypt 2006.

16. **Оперативная** экономика [Электронный ресурс]. — [2007]. — Режим доступа: http://www.opec.ru/print.aspx.

УДК 621.322

А. В. Суханов, канд. техн. наук,
нач. управления, ЗАО "Эврика",
Санкт-Петербург, e-mail: AVSuhanov@eureca.ru

Подход к построению защищенных информационных систем

С позиций биосистемной аналогии рассмотрены методологические вопросы проектирования сложных кибернетических систем, к которым в полной мере относятся средства мониторинга безопасности информационных систем (ИС). Предложен подход к построению защищенных ИС, определяющий основные положения и методологию создания защищенных интеллектуальных ИС. Подход основан на аналогии архитектуры и механизмов защиты биологических систем и сложных кибернетических систем.

Ключевые слова: *биосистемная аналогия, защита информации, интеллектуальные системы защиты, базы знаний, адаптивные классификаторы.*

Применение интеллектуальных средств для целей защиты информационных систем (ИС) является характерной чертой текущего этапа эволюции информационных технологий (ИТ) [1, 2]. Основное внимание исследователей и разработчиков систем защиты направлено на обнаружение и оперативную нейтрализацию последствий сетевых атак [3—5], а также обучение в режиме *on-line* интеллектуальных средств защиты для выявления несанкционированных действий в телекоммуникационных сетях и ИС [6, 7].

Актуальность проектирования защищенных ИС обусловлена высокими темпами развития, усложнением инфраструктуры и расширением функциональных возможностей ИТ. Прослеживается параллель между эволюцией видов биосистем и процессами развития современных ИТ [8]. Биосистемы развиваются благодаря совершенной защите информационных процессов, а дальнейшее развитие ИТ связано с обеспечением защищенности ИС, адекватной росту сложности информационных технологий. Перспективным методом разработки систем информационной безопасности (СИБ) является использование в искусственных системах аналогии с механизмами защиты (МЗ) информационных процессов и ресурсов, характерных для биосистем.

Работа посвящена методологическим вопросам проектирования сложных кибернетических систем, поставленных с позиции биосистемной аналогии [9]. Предложено проектирование ИС и средств обеспечения безопасности ИС осуществлять как единый процесс построения иерархической адаптивной системы с внутренне присущим свойством "защищенность". Процесс проектирования предполагается начинать с выбора надежной элементной базы, соответствующей требованиям функциональной устойчивости, алгоритмической универсальности и защищенности. Согласно принципу биосистемной аналогии с уровня элементной базы следует применять дублирование и избыточное кодирование информации, что свойственно элементному базису нейронных сетей (НС).

По аналогии с биосистемами при проектировании ИС следует осуществлять программную настройку нейросетевых базовых блоков, в процессе которой:

- в базовых блоках формируется набор взаимосвязанных интерфейсом функциональных устройств (аналогов органов), оговоренных в спецификации на проектирование и выполненных на основе формальных нейронов (ФН) — аналогов клеток;
- обмен информацией между функциональными устройствами организуется через интерфейс в виде закодированных сообщений;
- в процессе создания устройств в базовых блоках формируются адаптивные информационные поля НС, соответствующие функциям отдельных устройств ИС и интеллектуальных средств защиты;
- интеллектуальные средства защиты имеют иерархическую структуру;
- иммунная защита нижнего иерархического уровня осуществляет проверку сообщений, передаваемых по интерфейсу, по критерию "свой/чужой";
- защита верхнего иерархического уровня служит для накопления опыта нейтрализации механизмами защиты множества известных угроз.

Функциональная ориентация устройств осуществляется настройкой межнейронных связей НС, записи в локальную память базовых блоков системной информации в виде адаптивных информационных полей НС. Функции хранения системной информации (долговременная па-

мять), обработки и записи/считывания данных (оперативная память) должны быть разнесены для исключения несанкционированного изменения системной информации.

В процессе эксплуатации функции как отдельных устройств, так и ИС в целом могут изменяться в режиме адаптации следующим образом:

- добавление функции в информационную систему осуществляется аналогично процедуре формирования дополнительного устройства;
- изменение имеющихся функций связано с коррекцией в долговременной памяти системной информации соответствующего устройства, т. е. адаптацией информационного поля конкретной НС;
- адаптация информационных полей НС ассоциируется с процессом роста биосистемы, так как при изменении или добавлении функции информационной системы могут выделяться дополнительные ФН и происходить их интеграция в систему; при этом наблюдается естественное сочетание свойств стабильности (сохранение информации) и пластичности (настройка параметров ФН).

Функции защиты ИС реализуются описанным выше образом и корректируются в режиме адаптации НС при изменении множества угроз и наличии дестабилизирующих воздействий.

Основы методологии построения адаптивных средств защиты информации

Методология построения адаптивных средств защиты базируется на эволюционных свойствах нейронных сетей, связанных с адаптивностью, самообучением, возможностью представления опыта экспертов информационной безопасности (ИБ) в виде системы предикатных правил.

В процессе функционирования интеллектуальных средств защиты должна быть отражена последовательность выполнения следующих основных операций:

1) *классификация* угроз информационным ресурсам ИС — соотнесение выявленной угрозы с множеством известных угроз информационной безопасности (нижний уровень иерархии средств защиты);

2) *кластеризация* угроз информационным ресурсам ИС — как саморазвитие классификации при расширении множества угроз;

3) описание в виде *системы предикатных правил* соотношений "угрозы — механизмы защиты" (верхний уровень иерархии средств защиты);

4) *реализация* системы предикатных правил в виде специализированной нейросетевой структуры;

5) *адаптация* информационных полей НС (соответственно, и системы предикатных правил);

6) *анализ* структуры межнейронных связей информационных полей НС и "прозрачной" системы предикатных правил для выявления наиболее используемых механизмов защиты;

7) *формулирование* новых правил для *формирования* спецификации на разработку отсутствующих в ИС механизмов защиты.

Системный подход к моделированию интеллектуальных средств защиты

Системный подход обусловливает методологию, которой необходимо руководствоваться при разработке кибернетических систем. Базовыми принципами системного подхода являются [10] целеобусловленность, относительность, управляемость, связность и моделируемость.

Моделируемость служит основным средством разработки и верификации, позволяющим предотвратить ошибки проектирования кибернетических систем, к которым относятся системы защиты. В соответствии с принципом *связности* при разработке эффективных средств защиты ИС целесообразно рассматривать объект защиты комплексно, как составную часть сложной кибернетической системы, объединяющей в единой модели объект защиты, среду, средства защиты и угрозы злоумышленника как взаимосвязанные элементы [11].

Динамика множества угроз в процессе эксплуатации защищаемой ИС проявляется через новые уязвимости, не отраженные в исходной модели, и возникает потенциальная возможность реализации новых угроз безопасности информационным ресурсам и процессам. Поэтому целесообразно рассматривать модель средств защиты в динамике, начиная с начального этапа жизненного цикла системы, а нейросетевые средства, обладающие свойствами адаптивности и самообучения, в качестве базы для построения защищенных ИС.

Динамичный характер множества угроз выдвигает свойство *адаптивности* в разряд первоочередных качеств, необходимых средствам защиты. Не менее важным качеством является возможность реализации в ИС *накопленного опыта* нейтрализации угроз в информационных полях НС. Свойство *адаптивности* позволяет при ограниченных затратах на организацию средств защиты обеспечить заданный уровень безопасности ИС за счет оперативной реакции на изменение множества угроз.

Опыт средств защиты может храниться и передаваться в поколениях (модификация ИС) в виде распределенных адаптивных информационных полей: поля *известных угроз* на нижнем, иммун-

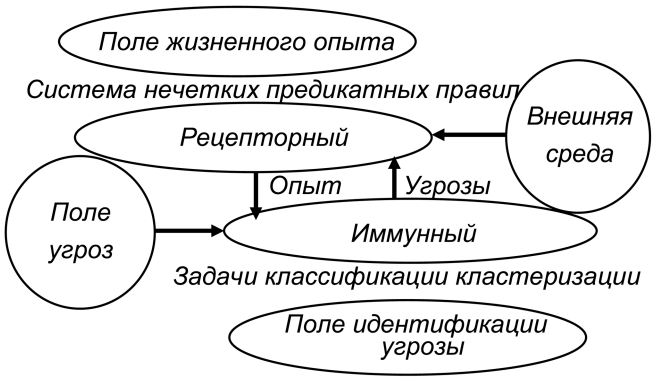


Рис. 1. Иерархия уровней интеллектуальных средств защиты

ном, уровне и поля *жизненного опыта* на верхнем, рецепторном, уровне средств защиты (рис. 1).

Процесс адаптации поля известных угроз связан с решением задач классификации, кластеризации. Изменение множества известных угроз отражается на верхнем уровне средств защиты в соответствующей модификации информационного поля жизненного опыта, реализованного в виде специализированной нейросетевой структуры, которая, в свою очередь, описывается системой предикатных правил. Процесс адаптации поля жизненного опыта связан с обучением НС, которое адекватно видоизменяет систему предикатных правил, ставящую в соответствие известным угрозам механизмы защиты.

Анализ взаимосвязанных пар "угроза—уязвимость" позволяет поставить в соответствие каждой угрозе, оговоренной в спецификации на проектирование защищаемой ИС, соответствующие уязвимости ИС. Экономически целесообразно закрыть механизмами защиты все выявленные уязвимости системы, а изменение множества угроз сопровождать процессом адаптации информационных полей известных угроз и жизненного опыта средств защиты.

Если в качестве базовой выбрать одну из многоуровневых моделей СИБ [12, 13], то вначале модель будет содержать минимальное число МЗ, достаточное для защиты выявленных уязвимостей ИС, которые будут пополняться при расширении множества угроз.

Методика проектирования адаптивных средств защиты информации

Пусть информационное поле нижнего уровня средств защиты обучено на всем поле известных угроз, т. е. возможна идентификация каждой из известных угроз, и нейросетевые средства защиты находятся в режиме работы. Пусть заданным угрозам x_i, $i = \overline{1, N}$, в процессе проектирования системы поставлены в соответствие выявленные уязвимости v_j, $j = \overline{1, J}$, и назначены МЗ z_k, $k = \overline{1, K}$. Механизмы защиты будем подразделять на активированные, потенциальные (известные, но еще не активированные МЗ) и отсутствующие (недоступные для использования в данной ИС). Без потери общности изложения ограничим поле угроз поступлением в систему "чужих" сообщений x_p, $p = \overline{1, P}$, где $P \geqslant N$.

1. При поступлении в интерфейс системы "чужого" сообщения x_p информационным полем нижнего уровня средств защиты будет идентифицирована угроза, если она принадлежит множеству известных угроз $\{x_p,\ p = \overline{1, P}\}$.

2. Если выявленная угроза соответствует подмножеству заданных угроз $\{x_i,\ i = \overline{1, N}\}$, то "чужое" сообщение изымается из процесса обработки и фиксируется статистика активации в информационных системах данной угрозы.

3. Если же выявленная в процессе классификации угроза не из подмножества $\{x_i,\ i = \overline{1, N}\}$, то выполняются действия по п. 2 и перестройка средств защиты. Под контролем администратора безопасности системы осуществляются перевод в режим адаптации и обучение НС иммунного уровня, соотнесение новой угрозы с выявленными или потенциальными уязвимостями, перевод в режим адаптации и обучение НС верхнего уровня для нейтрализации ранее неспецифицированной угрозы x_p имеющимися МЗ из множества $\{z_k\}$.

4. Если невозможна нейтрализация выявленной угрозы имеющимися МЗ, необходимо расширение множества $\{z_k\}$ за счет активации адекватных угрозе механизмов защиты. При этом происходит коррекция многоуровневой модели средств защиты путем активации ряда потенциальных МЗ информации и обучения НС верхнего уровня.

5. Если исчерпаны потенциальные МЗ и не получено соотнесение угрозы со средствами для ее нейтрализации, то под контролем администратора безопасности системы выполняется перевод нейросетевых средств защиты верхнего уровня в режим адаптации, расширение информационного поля НС верхнего уровня (введение дополнительного ФН) и обучение НС для нейтрализации ранее неспецифицированной угрозы x_p отсутствующими МЗ информации. В последнем случае анализ обученной НС верхнего уровня позволяет сформулировать систему требований к отсутствующим в системе МЗ.

Перестройка многоуровневой модели средств защиты может быть реализована с привлечением механизма нечеткого логического вывода и архитектурных решений нейронечетких сетей и сетей адаптивного резонанса [14—16].

Механизмы реализации адаптивных свойств средств защиты информации

Основными механизмами реализации *адаптивных свойств* средств защиты следует считать: способность распределенного информационного поля НС к *накоплению знаний* в процессе обучения; механизм *логического вывода*, который позволяет представить опыт экспертов в области защиты информации в виде системы предикатных правил и использовать его для *предварительного обучения* НС; способность нейросетевых средств к *классификации и кластеризации*.

Логический вывод. Нечеткое отношение $R = A \to B$ отражает знания эксперта $A \to B$ в виде причинного отношения посылки (угрозы) и заключения (механизма защиты), где операция $\to$ соответствует нечеткой импликации. Отношение R можно рассматривать как нечеткое подмножество прямого произведения $X \times Y$ полного множества угроз X и механизмов защиты Y, а процесс получения нечеткого результата вывода B' по посылке A' и знаниям $A \to B$ — в виде композиционного правила: $B' = A' \bullet R = A' \bullet (A \to B)$, где $\bullet$ — операция, например, max-min-композиции.

Механизм логического вывода основан на базе знаний, формируемой специалистами предметной области в виде системы предикатных правил вида:

Π_1 : если x есть A_1, то y есть B_1;
Π_2 : если x есть A_2, то y есть B_2;
...
Π_n : если x есть A_n, то y есть B_n,

где x и y соответственно входная переменная (например угроза) и переменная вывода (к примеру, механизм защиты), а A_i и B_i — функции принадлежности непрерывных переменных (НП).

Логический вывод, как правило, включает следующие этапы [15].

1. *Введение нечеткости*: по функциям принадлежности, заданным на области определения входных НП, исходя из фактических значений НП, назначается степень истинности каждой угрозы для каждого правила.

2. *Логический вывод*: по степени истинности угроз формируются заключения по каждому из правил, образующие нечеткое подмножество для каждого МЗ.

3. *Композиция*: нечеткие подмножества для каждого МЗ объединяются в целях формирования нечеткого подмножества для всех МЗ (по всем правилам).

4. *Приведение к четкости*: сводится к преобразованию нечеткого набора выводов по всем правилам в четкое значение итоговой защищенности системы.

Нейросетевая классификация и кластеризация в адаптивных средствах защиты могут быть реализованы с использованием нечетких НС или НС адаптивного резонанса [16, 17].

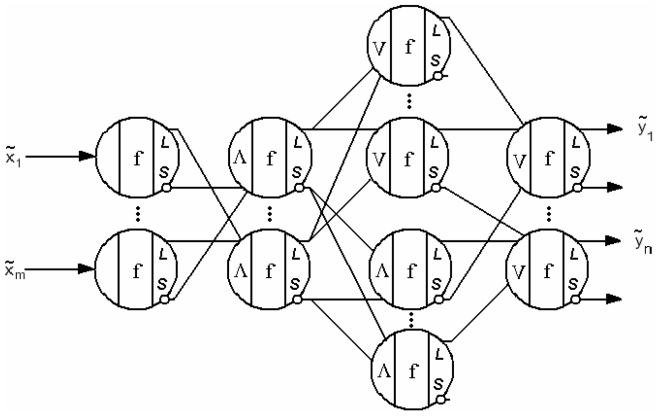


Рис. 2. Нейронечеткий классификатор

Нейронечеткие сети (рис. 2) [18] используют механизм нечеткого логического вывода и базу знаний, формируемую экспертами ИБ в виде системы правил:

Π_1: если $\tilde{x}_1$ есть A_{11} и ... $\tilde{x}_n$ есть A_{1n},
то $\tilde{y} = B_1$;
Π_2: если $\tilde{x}_1$ есть A_{21} и ... $\tilde{x}_n$ есть A_{2n},
то $\tilde{y} = B_2$;
.....
Π_k: если $\tilde{x}_1$, есть A_{k1} и ... $\tilde{x}_n$ есть A_{kn},
то $\tilde{y} = B_k$,

где $\tilde{x}$ и $\tilde{y}$ — нечеткие входная переменная и переменная вывода; A_{ij} и B_i, $i = \overline{1,k}$, $j = \overline{1,n}$, — соответствующие функции принадлежности.

При реализации системы предикатных правил в топологии нейронечеткой сети находят отражение следующие этапы нечеткого логического вывода:

- *введение нечеткости* — по функциям принадлежности, заданным на области определения посылок, исходя из фактических значений нечетких переменных $\tilde{x}_i$, определять степень истинности каждой посылки;
- *логический вывод* — по степени истинности посылок формировать заключения по каждому из правил, образующие нечеткое подмножество для каждой переменной вывода по каждому из правил;
- *композиция* — полученные на предыдущем этапе нечеткие подмножества для каждой переменной вывода по всем правилам объединять в целях формирования нечеткого подмножества для всех переменных вывода.

Сети теории адаптивного резонанса (*Adaptive Resonance Theory Network*, ART) [16] применяются для кластеризации многомерных векторов. Сети ART имеют множество модификаций, но интерес

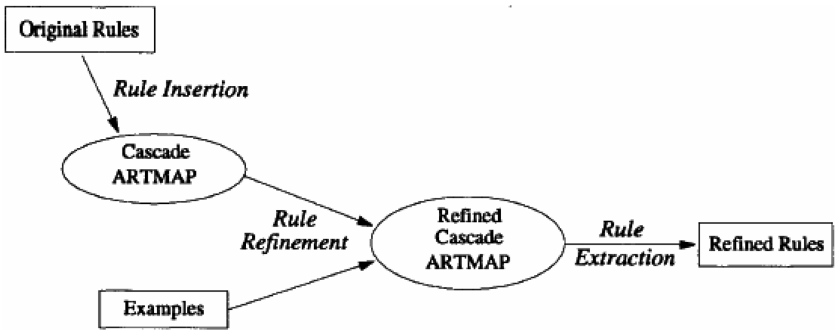


Рис. 3. Сеть Cascade ARTMAP, использующая априорные знания

для дальнейших исследований представляет Cascade ARTMAP (рис. 3) [19], которая позволяет включать в информационное поле ART-сети априорное знание, представленное в виде системы предикатных правил.

Наличие исходной базы знаний не только позволяет повысить эффективность обучения НС, но и дополнить информационное поле НС знаниями, отсутствующими в обучающих примерах. Причем неполные или частично достоверные правила могут быть откорректированы нейронной сетью в процессе обучения.

Используя алгоритм извлечения правил, информационное поле обученной НС может быть преобразовано обратно в систему предикатных правил, что позволяет сравнить исходные правила и модифицированную сетью базу знаний. Результаты экспериментов показали, что априорное знание увеличивает точность классификации, особенно при ограниченном наборе обучающих примеров [19].

Накопление опыта в интеллектуальных средствах защиты происходит в информационных полях НС в процессе обучения.

Изначально в средствах защиты формируется система предикатных правил для всех известных МЗ $\{z_k, k = \overline{1, K}\}$, так же, как нейросетевые средства идентификации угроз обучены на всем поле известных угроз $\{x_p, p = \overline{1, P}\}$. Незаданным угрозам во входном векторе x соответствуют нулевые значения координат, а деактивированным МЗ — близкие к нулю значения степени использования данного механизма защиты в формировании значения итоговой защищенности системы.

Задавая пороговые значения для величин z_k, $k = \overline{1, K}$, можно определять как наименее задействованные, так и наиболее эффективно используемые механизмы в обеспечении безопасности защищаемой системы.

После активации всех потенциальных механизмов защиты информации и введения дополнительных ФН в последний скрытый слой НС, соответствующий размерности вектора известных механизмов защиты, происходит *расширение системы предикатных правил*. Таким образом, средства защиты самостоятельно формируют правило, описывающее отсутствующий МЗ в защищаемой ИС. При последующей адаптации произойдет обучение нейронных сетей под отсутствующий МЗ, направленный на нейтрализацию ранее неспецифицированной угрозы x_p. Анализ дополнительного предикатного правила позволяет сформировать спецификацию на проектирование отсутствующего в системе механизма защиты.

Список литературы

1. **Гриняев С. Н.** Интеллектуальное противодействие информационному оружию. М.: СИНТЕГ, 1999.
2. **Tambe M., Pynadath D. V.** Towards Heterogeneous Agent Teams // Lecture Notes in Artificial Intelligence. V. 2086, Springer Verlag, 2001.
3. **Бочков М. В.** Реализация методов обнаружения программных атак и противодействия программному подавлению в компьютерных сетях на основе нейронных сетей и генетических алгоритмов оптимизации // Сб. докл. VI Международной конф. SCM'2003. — СПб.: Изд. СПГЭТУ, 2003. Т. 1. С. 376—378.
4. **Норткатт С.** Анализ типовых нарушений безопасности в сетях. М.: Издательский дом "Вильямс", 2001.
5. **Noureldien A. N.** Protecting Web Servers from DoS/DDoS Flooding Attacks. A Technical Overview // Proc. of Iniernational Conference on Web-Management for International Organisations. Geneva. 2002. October.
6. **Городецкий В. И., Карсаев О. В., Котенко И. В.** Программный прототип многоагентной системы обнаружения вторжений в компьютерные сети // Труды конгресса "Искусственный интеллект в XXI веке". ICAI'2001. Т. 1. М.: Физматлит, 2001.
7. **Городецкий В. И., Котенко И. В.** Командная работа агентов-хакеров: применение многоагентной технологии для моделирования распределенных атак на компьютерные сети // Труды VIII конф. по искусственному интеллекту. КИИ-2002. — М.: Физматлит, 2002.
8. **Осовецкий Л. Г.** Научно-технические предпосылки роста роли защиты информации в современных информационных технологиях // Изв. вузов. Приборостроение. 2003. Т. 46. № 7. С. 5—18.
9. **Осовецкий Л. Г., Нестерук Г. Ф., Бормотов В. М.** К вопросу иммунологии сложных информационных систем // Изв. вузов. Приборостроение. 2003. Т. 46. № 7. С. 34—40.
10. **Красносельский Н. И., Воронцов Ю. А., Аппак М. А.** Автоматизированные системы управления в связи: Учебник для вузов. М.: Радио и связь, 1988. 272 с.
11. **Вихорев С. В., Кобцев Р. Ю.** Как узнать — откуда напасть или откуда исходит угроза безопасности информации // Защита информации. Конфидент. 2002. № 2.
12. **Осовецкий Л., Шевченко В.** Оценка защищенности сетей и систем // Экспресс-электроника. 2002. № 2—3. С. 20—24.
13. **Мельников В. В.** Защита информации в компьютерных системах. — М.: Финансы и статистика; 1997. — 368 с.
14. **Fuller R.** Neural Fuzzy Systems. — Abo: Abo Akademi University, 1995.
15. **Круглов В. В., Борисов В. В.** Искусственные нейронные сети. Теория и практика. — 2-е изд., стереотип. — М.: Горячая линия — Телеком, 2002.
16. **Carpenter G. A., Grossberg S., Markuzon N., Reynolds J. H., Rosen D. B.** Fuzzy ARTMAP: An adaptive resonance architecture for incremental learning of analog maps. // Froc. of the International Joint Conference on Neural Network. 1992.
17. **Neznevitsky M.** Artificial intelligence: a guide to intelligent systems. Addison-Wesley, 2002.
18. **Нестерук Ф. Г., Молдовян А. А., Нестерук Г. Ф., Нестерук Л. Г.** Квазилогические нейронечеткие сети для решения задач классификации в системах защиты информации // Вопросы защиты информации. 2007. № 1. С. 23—31.

УДК 004.057.4

В. Г. Найденко, канд. физ.-мат. наук,
ст. науч. сотр.,
Институт математики,
Национальная академия наук Беларуси,
г. Минск

О неявной аутентификации пользователей компьютерных сетей

Приводится анализ известного протокола распределения общего ключа с неявной аутентификацией сторон. Доказано, что этот протокол, предполагавшийся ранее безопасным, не является стойким. Даны рекомендации по совершенствованию и уточнению этого протокола.

Ключевые слова: *протоколы распределения общего ключа, аутентификация, анализ стойкости.*

Введение

Чтобы установить безопасное сообщение между пользователями компьютерной сети, необходимо наличие протоколов аутентификации и распределения ключей. Под *протоколом* обычно понимают *распределенный алгоритм*, т. е. совокупность алгоритмов для каждой стороны протокола, форматы сообщений, пересылаемых между сторонами, способ синхронизации действий сторон, описание действий при сбоях. Интуитивно протокол называется *безопасным*, или *стойким*, в котором законные стороны достигают своей цели, а злоумышленник — нет [1].

В основе протокола аутентификации содержится некоторый алгоритм проверки того факта, что идентифицируемый объект (сторона протокола, пользователь компьютерной сети) знает некоторую конфиденциальную информацию, причем метод проверки является косвенным, без предъявления этой информации [2].

Протокол аутентификации пользователей обычно совмещается с протоколом распределения ключей по каналу связи, т. е. происходит аутентифицированный обмен информацией между сторонами, который завершается формированием общего ключа, известного только данным сторонам протокола.

Протоколы распределения ключей с взаимной аутентификацией сторон можно разделить на два класса:

- протоколы с явной аутентификацией;
- протоколы с неявной аутентификацией.

Протоколы с явной аутентификацией предусматривают обмен сообщениями между сторонами, при котором стороны в явном виде доказывают свое знание общего выработанного ключа. В то же время в протоколах с неявной аутентификацией стороны формируют общий ключ без дополнительных доказательств знания этого ключа. Предполагается, что никто, кроме данных сторон, не знает сформированный ими общий ключ. Поэтому стороны могут далее безопасно обмениваться сообщениями, зашифрованными таким ключом. Следовательно, успешная расшифровка сообщений неявно подтверждает аутентификацию сторон протокола. Протоколы с неявной аутентификацией в вычислительном плане более эффективны по сравнению с протоколами с явной аутентификацией, однако, как мы далее увидим, они являются менее стойкими.

Любое отдельное выполнение протокола называется *сессией*. Термин "сессия" часто связывается с каким-нибудь определенным понятием, относящимся к отдельному выполнению протокола. Например, общий ключ, согласованный в ходе выполнения протокола, называется *сессионным ключом*, или ключом сессии. Как правило, сессия протокола состоит из последовательности сообщений между сторонами и описывается с помощью общепринятой системы обозначений [3].

Стороны протокола обычно обозначаются буквами латинского алфавита, например i, j и k. Последовательность сообщений:

1) $i \to j : M_1$;
2) $j \to k : M_2$;
3) $k \to i : M_3$

обозначает протокол, в котором сторона i посылает стороне j сообщение M_1, затем сторона j посылает сообщение M_2 стороне k, которая, в свою очередь, пересылает сообщение M_3 стороне i.

Атаки на протоколы проводятся *злоумышленником*, пытающимся нарушить работу протокола. Мы будем обозначать такого злоумышленника символом E. Тогда $E(i)$ представляет собой злоумышленника E, действующего в роли законной стороны i. Обычно предполагается, что злоумышленник $E(i)$ может просматривать все сообщения законных сторон протокола и заменять одни сообщения другими, объявляя, что они посланы стороной i. Злоумышленник $E(i)$ также может перехватывать сообщения, предназначенные для стороны i, и, при желании, удалять их.

Сообщения могут быть зашифрованы. Через $[M]_K$ будем обозначать сообщение M, зашифрованное симметричным ключом K.

Оценка протокола распределения общего ключа с неявной взаимной аутентификацией сторон

Формальное описание протокола. Проведем анализ стойкости протокола 4 из работы [4], предна-

значенного для формирования общего ключа с неявной взаимной аутентификацией сторон i и j. В статье [4] формулируется гипотеза о том, что данный протокол удовлетворяет необходимым требованиям стойкости в модели со случайными оракулами. Модель со случайными оракулами основана на идеализированном представлении примитивов (базовых понятий) протокола как случайных оракулов [5]. Кратко идея этого подхода заключается в следующем. Чтобы упростить доказательство стойкости протокола некоторые примитивы (например, хэш-функции) заменяются датчиками случайных чисел (так называемыми случайными оракулами), к которым имеют доступ все стороны протокола (в том числе и возможный злоумышленник). Затем в этой упрощенной модели доказывается стойкость рассматриваемого протокола и делается предположение, что при обратной замене случайных оракулов на реальные примитивы исходный протокол также останется стойким. Необходимо лишь быть уверенным, что эти примитивы достаточно стойки сами по себе, в обычном смысле.

Хотя подтверждения гипотезы о сохранении стойкости практического протокола до сих пор не предъявлено ни авторами метода, ни криптографами-практиками, еще ни один протокол, для которого заявленная стойкость была доказана в модели со случайными оракулами, не был взломан (см., например, [6]).

Однако мы покажем, что рассматриваемый протокол не является достаточно стойким. Здесь и далее нотация α^x будет обозначать возведение величины α в степень x по модулю некоторого публично известного большого простого числа p. Величина α — публично известное натуральное число, генератор группы Диффи—Хеллмана. Через H обозначим публично известную хэш-функцию. Исходными данными протокола являются следующие параметры.

Для стороны i:

- секретный долговременный ключ S_i стороны i — случайное целое число, сгенерированное стороной i;
- открытый долговременный ключ стороны j, число $P_j = \alpha^{S_j}$.

Для стороны j:

- секретный долговременный ключ S_j — случайное целое число, сгенерированное стороной j;
- открытый долговременный ключ стороны i, число $P_i = \alpha^{S_i}$.

Протокол формирования общего ключа со взаимной аутентификацией сторон имеет следующую схему:

1) $i \to j : \alpha^{R_i}$;

1) $j \to i : \alpha^{R_j}$.

Описание действий сторон протокола:

- сторона i генерирует случайное число R_i, которое хранит в секрете, затем вычисляет значение α^{R_i} и пересылает его стороне j;
- сторона j генерирует случайное число R_j, которое хранит в секрете, затем вычисляет значение α^{R_j} и пересылает его стороне i.

Сторона i вычисляет сессионный ключ

$$K_{ij} = H((\alpha^{R_j})^{S_i}, (\alpha^{S_j})^{R_i}) = H(\alpha^{S_i R_j}, \alpha^{S_j R_i}).$$

Сторона j вычисляет сессионный ключ

$$K_{ji} = H((\alpha^{S_i})^{R_j}, (\alpha^{R_i})^{S_j}) = H(\alpha^{S_i R_j}, \alpha^{S_j R_i}).$$

Таким образом, стороны i и j вычислили общий ключ сессии $K_{ij} = K_{ji}$. Далее стороны протокола безопасно обмениваются сообщениями, зашифрованными этим ключом K_{ij}. Если стороны успешно расшифровывают эти сообщения, то аутентификация сторон подтверждается.

Нарушение неявной аутентификации. Допустим, что злоумышленник E знает некоторое натуральное число S_E, такое, что $\alpha^{S_E} = 1$. Пусть злоумышленник E ведет себя как законная сторона E и регистрирует в качестве своего открытого ключа величину 1 (единицу), а в качестве секретного ключа использует величину S_E. Тогда злоумышленник E может реализовать следующую схему атаки на протокол:

1) $i \to E(j) : \alpha^{R_i}$;

2) $E(i) \to j : 1$;

3) $j \to E(i) : \alpha^{R_j}$;

4) $j' \to E : \alpha^{R_{j'}}$;

5) $E(j) \to i : 1$;

6) $E \to i' : \alpha^{R_j}$;

7) $E \to j' : \alpha^{R_i}$;

8) $i' \to E : \alpha^{R_{i'}}$.

Здесь i' и j' — копии сторон i и j, участвующие в параллельных сессиях протокола (стороны i и j участвуют в сессии протокола между собой, а i' и j' — в сессиях со стороной E).

Сторона протокола i вычисляет сессионный ключ $K_{ij} = H(1^{S_i}, (\alpha^{S_j})^{R_i}) = H(1, \alpha^{S_j R_i})$. В то же время сторона протокола j вычисляет сессионный ключ $K_{ji} = H((\alpha^{S_i})^{R_j}, 1^{S_j}) = H(\alpha^{S_i R_j}, 1)$.

Стороны i' и j' вычисляют сессионные ключи со стороной E как $K_{i'E} = H((\alpha^{R_j})^{S_i}, 1^{R_{i'}}) = H(\alpha^{S_i R_j}, 1)$ и $K_{j'E} = H(1^{R_{j'}}, (\alpha^{R_i})^{S_j}) = H(1, \alpha^{S_j R_i})$ соответственно.

Нетрудно видеть, что справедливы равенства $K_{ij} = K_{j'E}$ и $K_{ji} = K_{i'E}$. Злоумышленник E не может вычислить ключи $K_{i'E}$ и $K_{j'E}$, однако он может нарушить неявную аутентификацию законных сторон протокола, реализуя следующую атаку:

1) $i' \to E : [M_1]_{K_{i'E}}$;

2) $j' \to E : [M_2]_{K_{j'E}}$;

3) $E(i) \to j : [M_1]_{K_{i'E}}$;

4) $E(j) \to i : [M_2]_{K_{j'E}}$.

То есть, получая от сторон i' и j' зашифрованные сообщения $[M_1]_{K_{i'E}}$ и $[M_2]_{K_{j'E}}$, злоумышленник просто перенаправляет их сторонам j и i, но уже от имени сторон i и j соответственно. Так как $K_{ij} = K_{j'E}$ и $K_{ji} = K_{i'E}$, то сторона j может расшифровать сообщение $[M_1]_{K_{i'E}}$, а сторона i — сообщение $[M_2]_{K_{j'E}}$. Таким образом, законные стороны i и j "обменялись" сообщениями $[M_1]_{K_{i'E}}$ и $[M_2]_{K_{j'E}}$, которые они не посылали друг другу! Злоумышленнику нужно лишь заблокировать канал связи между i и j, чтобы они не получали друг от друга сообщения, и такую атаку будет невозможно распознать.

Вместе с тем злоумышленник E может реализовать еще одну атаку:

1) $i \to E(j) : [M_3]_{K_{ij}}$;

2) $j \to E(i) : [M_4]_{K_{ji}}$;

3) $E \to j' : [M_3]_{K_{ij}}$;

4) $E \to i' : [M_4]_{K_{ji}}$.

Перехватывая от сторон i и j зашифрованные сообщения $[M_3]_{K_{ij}}$ и $[M_4]_{K_{ji}}$, злоумышленник просто пересылает их сторонам j' и i' но уже от имени стороны E. Так как $K_{j'E} = K_{ij}$ и $K_{i'E} = K_{ji}$, то сторона j' может расшифровать сообщение $[M_3]_{K_{ij}}$, а сторона i' — сообщение $[M_4]_{K_{ji}}$. Таким образом, законные стороны j' и i' "получили" сообщения $[M_3]_{K_{ij}}$ и $[M_4]_{K_{ji}}$ от стороны E и успешно их расшифровали, хотя злоумышленник E не создавал эти сообщения и вообще не может их расшифровать!

Таким образом, злоумышленник E может нарушить неявную аутентификацию с помощью атаки "малые подгруппы" [7].

Потеря совершенной секретности. Рассмотрим здесь ситуацию, когда злоумышленнику становятся известны долговременные секретные ключи некоторых законных сторон протокола. Совершенная секретность [4] предусматривает, что теперь этот злоумышленник может замаскироваться под какую-нибудь законную сторону и генерировать новые сессионные ключи, однако он не в состоянии раскрыть старые сессионные ключи, созданные этой законной стороной. Покажем, что для протокола 4 взаимной аутентификации и распределения ключей [4] совершенная секретность не выполняется. То есть если злоумышленник знает долговременные ключи двух законных сторон протокола, то он без труда может восстановить все сессионные ключи, вычисленные этими сторонами. Пусть злоумышленник E пока просто подслушивает стороны i и j протокола, т. е. имеем следующую схему подслушивания для некоторой сессии:

1) $i \to E(j) : \alpha^{R_i}$;
2) $E(i) \to j : \alpha^{R_i}$;
3) $j \to E(i) : \alpha^{R_j}$;
4) $E(j) \to i : \alpha^{R_j}$.

Стороны протокола вычисляют общий сессионный ключ

$$K_{ij} = H(\alpha^{S_i R_j}, \alpha^{S_j R_i}).$$

Злоумышленник E сохраняет значения α^{R_i} и α^{R_j} и подслушивает все сообщения между сторонами i и j, зашифрованные сессионным ключом K_{ij}.

Естественно, злоумышленник E пока не может расшифровать эти сообщения. Допустим, что через какое-то время он завладел долговременными ключами S_i и S_j сторон i и j и тогда он восстанавливает старый ключ сессии

$$K_{ij} = H((\alpha^{R_j})^{S_i}, (\alpha^{R_i})^{S_j}) = H(\alpha^{S_i R_j}, \alpha^{S_j R_i})$$

и может расшифровывать записанные им ранее сообщения между сторонами i и j.

Рекомендации по совершенствованию протокола

В данном разделе мы дадим рекомендации по повышению стойкости протокола 4 взаимной аутентификации и формированию общего ключа [4].

Для того чтобы обеспечить стойкость протокола, необходимо использовать не просто удостоверенные открытые ключи сторон протокола, но и доказать, что они не подвержены атаке "малые подгруппы" [7]. По крайней мере, для всякой стороны i или j протокола взаимной аутентификации должны выполняться соотношения $P_i \neq 1$ и $P_j \neq 1$, где P_i и P_j — открытые ключи сторон i и j соответственно. Кроме того, можно надежно защититься от атаки, нарушающей неявную аутентификацию сторон i и j, если условиться о следующем формате сообщений, зашифрованных общим ключом сессии K_{ij}.

Каждое зашифрованное сообщение $[M]_{K_{ij}}$ должно содержать идентификаторы стороны-отправителя и стороны-получателя (или адреса отправителя и получателя сообщения). Например,

сообщение может иметь вид $[i, j, Text]_{K_{ij}}$, где i — идентификатор отправителя, j — идентификатор получателя, *Text* — текст сообщения, которое сторона i посылает стороне j. Если, например, сторона j получает сообщение $[j, E, Text]_{K_{ij}}$ якобы от стороны i, то j выясняет, что произошло неправомерное распределение ключей, так как идентификаторы отправителя j и получателя E в этом сообщении не совпадают со сторонами i и j. То есть сторона j может установить, что сообщение $[j, E, Text]_{K_{ij}}$ было направлено ей злоумышленником.

Кроме того, чтобы соблюдался такой атрибут стойкости, как совершенная секретность, можно использовать следующую формулу вычисления общего ключа:

$$K_{ij} = K_{ji} = H(\alpha^{S_i R_j}, \alpha^{S_j R_i}, \alpha^{R_i R_j}).$$

Предложенная выше формула является менее эффективной по времени вычисления, но обеспечивает большую стойкость протокола.

Список литературы

1. **Ященко В. В.** Введение в криптографию. М.: Издательство МЦНМО, 2003. 400 с.
2. **Сапегин Л. Н.** Типичные дефекты в криптографических протоколах // Спец. техника средств связи. Сер. Системы, сети и технич. средства конфиденц. связи. 1996. Вып. 1. С. 68—83.
3. **Clark J., Jacob J.** A survey of authentication protocol literature: version 1.0. // Internal Report, University of York. 1997. November. [http://www.wm.tue.nl/~ecss/downloads/clarkjacob.pdf]
4. **Blake-Wilson S., Johnson P., Menezes A.** Key agreement protocols and their security analysis // Proc. of the Sixth IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science. 1997. 1355. P. 30—45.
5. **Bellare M., Rogaway P.** Random oracles are practical: a paradigm for designing efficient protocols // Proc. of the 1st ACM Conference on Computer and Communications Security. 1993. P. 62—73.
6. **Аграновский А. В., Хади Р. А., Балакин А. В**. Эвристическое доказательство стойкости криптосистем // Сб. трудов научно-технической конференции "Безопасность информационных технологий". Т. 2. Пенза: Изд. Пензенского научно-исследовательского электротехнического института, 2001.
7. **Zuccherato R.** Methods for avoiding the "small-subgroup" attacks on the Piffle—Hellman key agreement method for S/MIME. RFC 2785, Network Working Group. March 2000. [http://rfc.sunsite.dk/rfc/rfc2785.html].

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ И БИОЛОГИИ

УДК 330.47:001.891.573

М. А. Кожевников[1], программист, **Ю. В. Марчук**[1], программист,
О. Н. Хамидулина[1], врач-ортопед, **А. И. Монтиле**[2], канд. физ.-мат. наук, доц.,
И. А. Погосян[1], д-р мед. наук, руководитель Областного детского ортопедического центра,
e-mail: pogosyan@bonum.info

[1] Государственное учреждение здравоохранения Свердловской области
Детская клиническая больница восстановительного лечения,
Научно-практический центр (ГУЗ СО ДКБВЛ НПЦ) "БОНУМ", Екатеринбург

[2] Уральский государственный лесотехнический университет (УГЛТУ), Екатеринбург

Разработка средства поддержки диагностики ортопедической патологии на основе дискриминантного анализа клинико-анамнестических данных

Описана разработка средства аналитической поддержки процесса дифференциальной диагностики патологии шейного отдела позвоночника (ШОП). Представлены технология выявления типа нарушения ШОП, а также механизм формирования структурной патологии верхнешейного отдела позвоночника. Разработаны и апробированы алгоритмы информационно-программной поддержки на основе адаптации дискриминантного анализа (ДА). Выявлена несостоятельность ДА как средства многомерного анализа данных в вопросах поддержки диагностики ортопедической патологии без предварительной его адаптации.

Ключевые слова: *дискриминантный анализ, диагностика, информационно-программная поддержка, набор решающих правил, разрешение противоречий, алгоритм дифференциальной диагностики, структурные нарушения шейного отдела позвоночника, краниовертебральная область.*

Актуальность темы

На сегодняшний день является актуальным вопрос новых постановок задач математического моделирования в медико-биологической практике. В большинстве своем ряд направлений, связанных с моделированием сложных медицинских и биологических систем, не получали развития, в частности, в травматологии-ортопедии.

Математическому моделированию медицинскою диагностики и применению моделирования для решения актуальных практических задач, а также проблемам в области разработки медицинских информационных сис-

тем поддержки диагностики посвящены работы, в основном касающиеся вопросов теоретического характера, строгих постановок задачи, применения математического анализа для разработки средств постановки диагноза [1—8].

К проблемам разработки средств поддержки медицинской деятельности и диагностики ортопедической патологии (ОП), в частности, относят: слабую формализованность знаний и высокую размерность пространства признаков, не позволяющих врачу (особенно молодому специалисту) сразу охватить полный набор диагностических маркеров патологии [9]. Особенностью диагностики ОП в раннем детском возрасте также является слабая информативность каждого из признаков, взятых в отдельности [10—13].

Приводимые в литературе алгоритмы донозологической диагностики на основе регрессионных моделей физиологических состояний исходят из представления о том, что вся шкала переходов от одного состояния к другому может быть описана линейной функцией. На самом деле сложные физиологические и патологические процессы адаптации организма к условиям окружающей среды вряд ли имеют линейную природу. Это обусловлено тем, что на разных стадиях адаптации взаимодействие биомеханических процессов, компенсации и собственно адаптационных механизмов складывается поразному. Пространства, в которых развертываются процессы компенсации различных отделов опорно-двигательного аппарата, крайне неоднородны. Поэтому для точного их описания следует использовать такие математические методы, как нелинейные регрессионные уравнения или полиномы различной степени [14, 15]. Согласно описаниям различных средств математического моделирования, используемых в медицинских приложениях, одним из наиболее подходящих методов, ориентированных на решение задач медицинской диагностики, является дискриминантный анализ [14—19].

Данная работа посвящена решению проблемы разработки специализированного механизма поддержки диагностики ортопедической патологии, а именно патологии шейного отдела позвоночника (ШОП), или, что то же, краниовертебральной области (КВ). Этот механизм включает алгоритмы принятия решений на основе адаптации дискриминантного анализа с разработкой программного средства.

Материалы и методы

Исследованы 108 детей в возрасте от 6 до 16 лет с клиническими признаками патологии краниовертебральной области и 13 детей условно здоровых (без патологии КВО). Все дети обследованы ортопедом, неврологом, окулистом, сурдологом. Учитывались возраст, жалобы пациента (четыре признака), данные анамнеза (10 признаков), результаты обследования смежными специалистами (два показателя), данные клинического осмотра (20 переменных). У всех детей для подтверждения диагноза была проведена функциональная рентгенография шейного отдела позвоночника. Варианты патологии КВО следующие: диспластические изменения КВО; травматические изменения КВО; травматические изменения на фоне аномалий КВО и синдрома соединительно-тканной дисплазии (ССтД); травматические изменения шейного отдела позвоночника на фоне ССтД; без патологии КВО.

По результатам рентгенологического исследования экспертами были условно выделены степени выраженности травматических и диспластических изменений шейного отдела позвоночника: 1 — отсутствие травмы/дисплазии; 2 — травма/дисплазия 1-й степени выраженности; 3 — травма/дисплазия 2-й степени выраженности.

Результаты и их обсуждение

В ходе исследований обнаружено, что по отдельности характеристики состояния пациентов имеют низкую информативность и не позволяют применять типовой вариант разработки решающего правила поддержки диагностики. На основании исследований, затрагивающих анализ специфики данных и знаний предметной области "детская травматология — ортопедия", получен пакет структурных и алгоритмических моделей, отражающих особенности поддержки диагностики нарушений краниовертебральной области у детей с применением дискриминантного анализа.

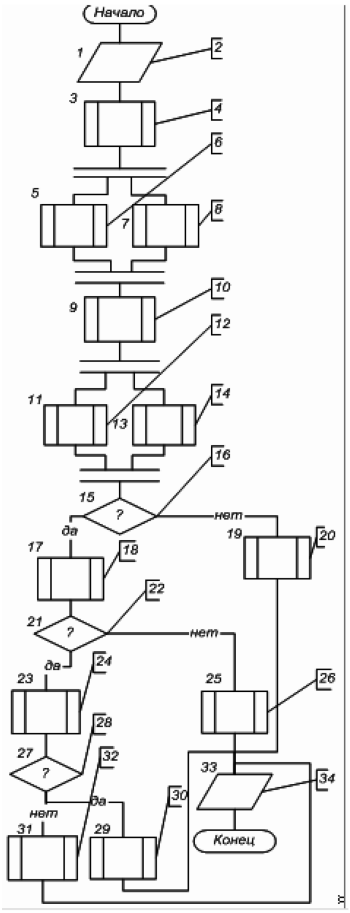
Авторами предложена балльная система оценки качественных показателей состояния опорно-двигательного аппарата. Предложена система оценки показателей, характеризующих жалобы пациента и его анамнез. Выбранные признаки использованы в качестве независимых переменных. Каждому признаку был присвоен числовой код в баллах по шкале оценки признаков (табл. 1). Для построения решающего механизма предложено применять набор специализированных решающих правил, использующих входную информацию по специальной двухуровневой схеме. Схема имеет в своей основе технологию постановки диагноза, отраженную в алгоритме дифференциальной диагностики (рис. 1). Основная особенность состоит в наличии нескольких вариантов сочетания типов патологии КВО.

Каждый тип патологии или их сочетание характеризуется своими маркерами и не может быть однозначно определен в силу встречаемости отдельных маркеров в нескольких вариантах развития патологического про-

Таблица 1

Клинические и анамнестические признаки у детей со структурной патологией КВО и различной степенью ее выраженности

№ п/п	Показатель	Принимаемые значения	Балльная шкала оценки
1	Возраст	Измеряется в годах	—
...	...	...	...
6	Недоношенность	Доношенный	1
		Недоношенность I—II степени	2
		Недоношенность III—IV степени	3
...	...	...	...
37	Степень отклонения пяточной кости	Норма	1
		I степень	2
		II степень	3



2. Жалобы пациента;
4. Анализ жалоб и сбор анамнеза пациента;
6. Проверить наличие маркеров травмы;
8. Проверить наличие маркеров дисплазии;
10. Провести клинический осмотр;
12. Проверить наличие клинических маркеров травмы;
14. Проверить наличие клинических маркеров дисплазии;
16. Есть ли проявления нарушений со стороны опорно-двигательного аппарата (ОДА);
18. Поставить предварительный диагноз;
20. Зафиксировать отсутствие патологии ШОП;
22. Есть ли сочетание маркёров травмы и дисплазии;
24. Анализировать результаты рентгенографии ШОП;
26. Поставить окончательный диагноз по преобладающей клинико-анамнестической картине;
28. Имеются ли проявления дисплазии на рентгенограмме ШОП;
30. Поставить диагноз "Травма КВО на фоне ССтД";
32. Поставить диагноз "Травма на фоне аномалии КВО и ССтД";
34. Диагноз, клинический статус, данные анамнеза, жалобы, результаты рентгенографии ШОП

Рис. 1. Алгоритм дифференциальной диагностики патологии краниовертебральной области у детей

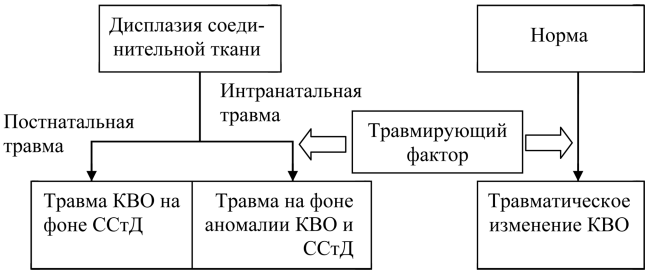


Рис. 2. Схема формирования структурных нарушений КВО с учетом механизма формирования патологии краниовертебральной области (интранатальный период — во время родов, постнатальный период — послеродовый период)

цесса. Кроме того, замечено, что механизм формирования структурной патологии краниовертебральной области обусловливает направленность формирования структурных нарушений КВО. Схема формирования структурных нарушений КВО с учетом механизма формирования патологии краниовертебральной области представлена на рис. 2.

На основании исследования особенностей диагностики нарушений КВО предложено использовать набор решающих правил — восемь двухуровневых структур, две из которых являются основными, шесть — вспомогательными. Принципиальная схема, используемая при построении решающих правил, представлена на рис. 3. Семантическое наполнение принципиальной схемы представлено в восьми вариантах в зависимости от ориентации решающего правила на выявление определенного подмножества альтернативных вариантов диагнозов.

Следующим этапом диагностики нарушений КВО является определение степени выраженности структурной патологии краниовертебральной области. Без предварительного определения типа структурной патологии КВО определение степени выраженности представляется невыполнимым с точки зрения предлагаемого решения.

Используя условное деление степеней выраженности травматических и диспластических изменений КВО на основании данных рентгенографии КВО, предложены две схемы распознавания степеней выраженности структурной патологии КВО (рис. 4, 5), дополняющие друг друга (для каждой вершины построено собственное решающее правило).

Класс А. Дети с диспластическими изменениями КВО (А1) или с травматическими изменениями в КВО (А2). **Класс Б.** Дети без патологии КВО (Б1); с травматическими изменениями на фоне аномалий КВО и синдрома соединительно-тканной дисплазии — смешанный тип (Б2); дети с травматическими изменениями шейного отдела позвоночника на фоне синдрома соединительно-тканной дисплазии — ССтД (Б3).

Класс А′. Дети с диспластическими изменениями КВО (А′1), или с травматическими изменениями в КВО (А′2), или без патологии КВО (А′3). **Класс Б′:** дети с

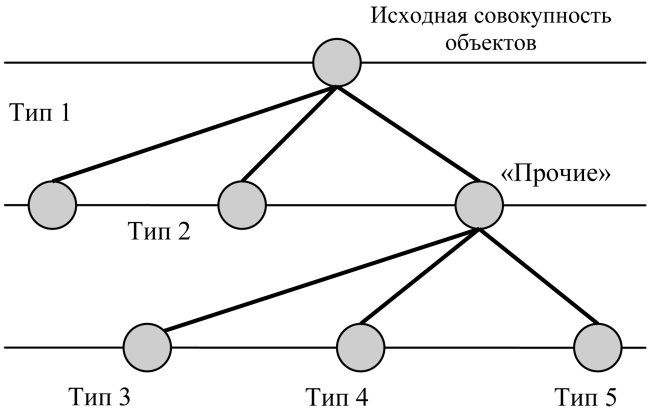


Рис. 3. Принципиальная схема решающих правил

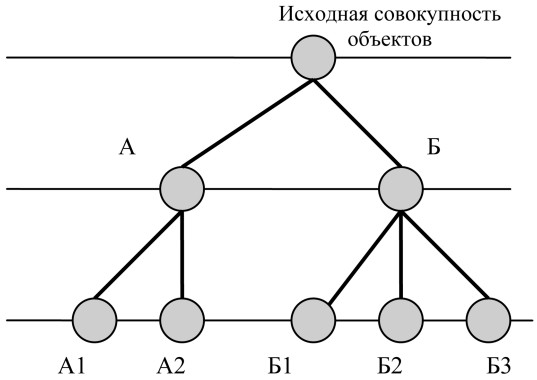


Рис. 4. Схема распознавания степеней выраженности структурной патологии КВО (первый вариант)

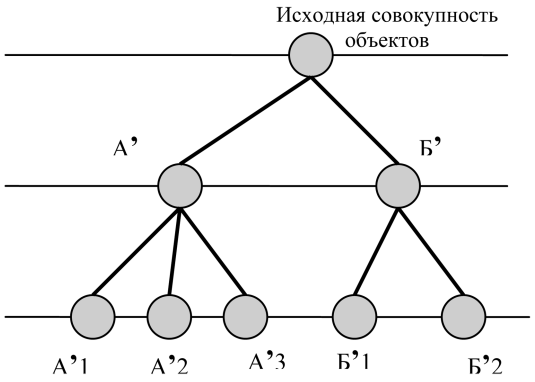


Рис. 5. Схема распознавания степеней выраженности структурной патологии КВО (второй вариант)

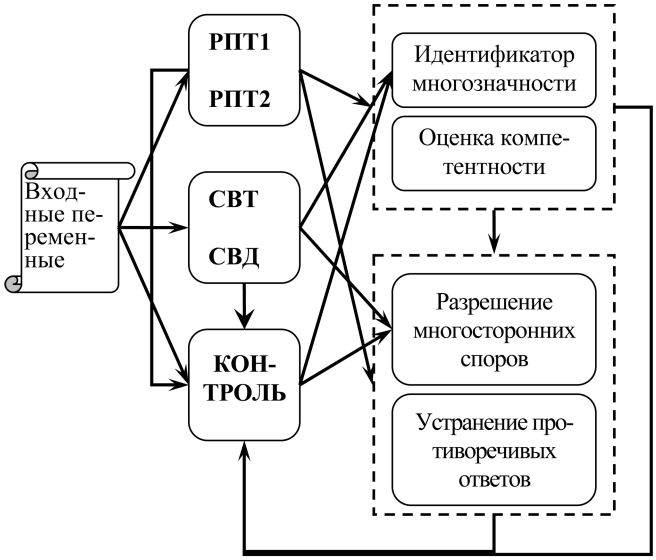


Рис. 6. Схема взаимодействия набора решающих правил в рамках пакета алгоритмов поддержки информационно-программной поддержки диагностики ортопедической патологии (СВТ — степень выраженности травмы, СВД — степень выраженности дисплазии)

травматическими изменениями на фоне аномалий КВО и синдрома соединительно-тканной дисплазии — (Б′1); дети с травматическими изменениями шейного отдела позвоночника на фоне синдрома соединительно-тканной дисплазии — ССтД (Б′2).

Предлагаемая авторами схема взаимодействия набора решающих правил подразумевает итерационный процесс, включающий механизмы обратной связи и самоконтроля (рис. 6). Предполагается автоматическая корректировка ответов в зависимости от сочетания ряда факторов, таких как степень удаленности исследуемого объекта от стационарных точек — центров групп, компетентность, чувствительность, специфичность и валидность решающих правил поддержки диагностики.

На первом этапе применяются два основных решающих правила (РПТ1 и РПТ2). Выявление типа патологии КВО осуществляется на всех предусмотренных структурой правил уровнях. Остальные диагностические правила (РПТ3, РПТ4, РПТ5, РПТ6, РПТ7, РПТ8) выступают в роли средства разрешения возможных противоречий в ответах первых двух (табл. 2). При этом противоречия разрешаются на основе канонических линейных дискриминантных функций второго уровня распознавания соответствующего вспомогательного решающего правила.

Схема также подразумевает использование алгоритмов выявления и разрешения споров в ответах решающего правила. Используются дублирующие друг друга алгоритмы и правила.

Под спорной ситуацией понимается одновременное нахождение исследуемого объекта в двух и более секторах — классах. Другими словами, объект принадлежит пересечению двух и более множеств объектов обучающей выборки. Авторами предложен алгоритм выявления подобных ситуаций. После разрешения спорных вопросов предполагается получение двух вариантов ответов (по одному от каждого из основных правил). При этом если ответы различные, тогда имеет место противоречие в ответах двух основных РП и задействуется схема разрешения противоречий с применением понятий чувствительности, специфичности, валидности диагностического теста [20] на основе применения соответствующего вспомогательного решающего правила, согласно табл. 2.

После определения типа структурной патологии КВО алгоритм предполагает выявление степени выраженности патологии КВО (по табл. 3).

После выявления типа и степени выраженности патологии КВО задействуется дополнительное решающее правило с функцией контроля (см. рис. 6), являющееся средством проверки корректности полученных ответов.

Заключение

Алгоритмы адаптации дискриминантного анализа к медицинской диагностике позволили получить более точный и адекватный инструмент выявления типа и сте-

Таблица 2

Матрица разрешения противоречий в ответах основных решающих правил

Ответы РПТ1/РПТ2	Дисплазия КВО	Травма КВО	Травма на фоне аномалии КВО и ССтД	Травма КВО на фоне ССтД
Дисплазия КВО	—	РПТ8/2	РПТ5/2	РПТ7/2
Травма КВО	РПТ8/2	—	РПТ6/2	РПТ4/2
Травма на фоне аномалии КВО и ССтД	РПТ5/2	РПТ6/2	—	РПТ3/2
Травма КВО на фоне ССтД	РПТ7/2	РПТ4/2	РПТ3/2	—

Таблица 3

Схема определения класса функций распознавания степени выраженности патологии КВО (основной и альтернативный наборы)

Тип патологии	Основной набор	Альтернативный набор
Без патологии КВО	Б	А'
Дисплазия КВО	А	А'
Травма КВО	А	А'
Травма на фоне аномалии КВО и ССтД	Б	Б'
Травма КВО на фоне ССтД	Б	Б'

пени выраженности патологии, при этом появилась возможность различать варианты сочетания патологических процессов.

Модель процесса диагностики дополнена алгоритмом адаптации метода многомерного анализа данных — дискриминантного анализа, используемым для учета возможного ошибочного ответа со стороны подмножества множества решающих правил. Алгоритм заключается в многократном уточнении диагноза и корректировке окончательного ответа за счет исключения из рассмотрения наименее правдоподобных вариантов.

Особенность медицинского знания и специфика данных предметной области "травматология-ортопедия" не позволяют применять дискриминантный анализ к построению средства поддержки диагностики ортопедической патологии по известным трафаретным схемам. Совокупность решающих правил поддержки диагностики способна обеспечить требуемый уровень качества решений при условии отражения каждым из них определенной стороны патологического процесса.

Дискриминантный анализ может служить средством построения решающих механизмов в задаче поддержки диагностики, однако полностью избежать ошибок без использования специальной логической последовательности применения диагностических правил не представляется возможным.

Разработанная на основе результатов исследования информационно-интеллектуальная система поддержки диагностики (свидетельство об официальной регистрации программы для ЭВМ № 2007614539) позволяет учесть особенности предметной области и повысить качество диагностических мероприятий. Испытания, проводимые на базе научно-практического центра "Бонум" (г. Екатеринбург), показали обнадеживающие результаты, а именно повышение точности выявления патологии в 1,6 раза по сравнению с типовым вариантом применения многомерного анализа данных.

Список литературы

1. **Бейли Н.** Математика в биологии и медицине. М.: Мир, 1970. 326 с.
2. **Беллман Р.** Математические методы в медицине. М.: Мир, 1987. 200 с.
3. **Казанцев В. С.** Математические методы и новые информационные технологии в решении медицинских задач: лекции. Екатеринбург: НПЦ Уралмедсоцэкономпроблем, 2002. 79 с.
4. **Кудрин А. Н., Пономарева Г. Т.** Применение математики в экспериментальной и клинической медицине. М.: Медицина, 1967. 356 с.
5. **Лапко А. В.** Статистические методы моделирования и принятия решений в развивающихся медико-биологических системах. Новосибирск, 1991. 223 с.
6. **Мазуров В. Д.** О плохо формализуемых задачах анализа сложных систем. // Математическое моделирование процессов в медицинских и биологических системах. Свердловск: УНЦ АН СССР, 1982. С. 3—7.
7. **Смит Дж.** Математические идеи в биологии. М.: Мир, 1970. 179 с.
8. **Штейн Л. Б.** Опыт прогнозирования с помощью ЭВМ. Л: Ленинградский институт, 1984. 146 с.
9. **Гридин В. Н., Тарасова О. Б.** Построение интеллектуальных диагностических систем в медицинских приложениях // Информационные технологии. 2007. № 7. С. 54—58.
10. **Афанасьев А. П., Овечкина А. В., Садофьева В. И.** Особенности течения диспластического сколиоза начальных степеней у детей в условиях промышленного города // Лечение и реабилитация детей-инвалидов с ортопедической и отоневрологической патологией на этапах медицинской помощи. СПб., 1997. — С. 73—74.
11. **Ветрилэ С. Т., Колесов С. В.** Аномалии развития и дисплазии верхнешейного отдела позвоночника (клиника, диагностика и лечение) // Вестник ин-та травматологии и ортопедии им. Н. Н. Приорова. 1997. № 1. С. 62—68.
12. **Мажейко Л. И.** Вертеброневрологические аспекты поражений краниовертебральной области у детей (клиника, диагностика): дис. ... канд. мед. наук: 14.00.13. Екатеринбург. 1997. — 157 с.
13. **Михайлов М. К.** Рентгенодиагностика родовых повреждений позвоночника. М.: ГОЭТАР-МЕД, 2001. 176 с.
14. **Клекка У. Р., Ким Дж.-О., Мьюллер Ч. У. и др.** Факторный, дискриминантный и кластерный анализ / Пер. с англ. А. М. Хотинского, С. Б. Королева, под ред. И. С. Енюкова. М.: Финансы и статистика, 1989. 215 с.
15. **Охтилев М. Ю., Соколов Б. В., Юсупов Р. М.** Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических объектов. М.: Наука, 2006. 410 с.
16. **Боровиков В. П., Боровиков И. П.** STATISTICA. Статистический анализ и обработка данных в среде Windows. М.: Филинъ, 1997. 608 с.
17. **Дюк В., Эмануэль В.** Информационные технологии в медико-биологических исследованиях. СПб.: Питер, 2003. 528 с.
18. **Каримов Р. Н.** Основы дискриминантного анализа: учебно-методическое пособие. Саратов: Изд. СГТУ, 2002. 108 с.
19. **Юнкеров В. И., Григорьев С. Г.** Математико-статистическая обработка данных медицинских исследований. СПб.: Изд. Военно-медицинской академии, 2002. 267 с.
20. **Кельмансон А. В.** Принципы доказательной педиатрии. — СПб.: Фолиант, 2004. 240 с.

УДК 004.9

В. П. Дьяконов, д-р техн. наук, проф., зав. каф.,
Ф. А. Хотова, аспирант,
Смоленский государственный университет

Матричная система MATLAB в биоинформатике

Рассматриваются возможности матричной системы MATLAB R2008a в биоинформатике. На примере анализа генной цепочки птичьего гриппа описана работа пакета расширения Bioinformatics Toolbox с мировыми Интернет-ресурсами по генетике, а также средства пакета по обработке микромассивов и данных масс-спектроскопии. Сделан вывод о возможности применения средств биоинформатики при решении различных научных и образовательных задач.

Ключевые слова: *биоинформатика, анализ генных цепочек, обработка микромассивов, масс-спектрометрические данные, кластеризация, филогенетические деревья.*

Введение

Биоинформатика — новое научное направление на стыке молекулярной биологии, генетики, математики и компьютерных технологий. Эта наука предполагает применение информационных технологий для изучения сложных биологических систем и применение биологических методов для эффективного решения задач в различных сферах науки и техники [1, 2]. Прежде всего это относится к хранению, систематизации, сравнительному анализу данных и информации, полученной в ходе молекулярных, геномных и протеомных исследований, и к моделированию отдельных процессов и взаимодействий в природе.

Представление о генах и генных цепочках

Информация о структуре живого организма и происходящих в нем процессах записана в особых информационных структурах организма — генах. Гены являются объектами только живой природы. Открытие генов — величайшее открытие в истории человечества. Оно позволило объяснить многие закономерности жизни живых организмов и целенаправленно влиять на них. Одна из задач генетики — клонирование живых организмов или их частей, т. е. их создание искусственным путем на основе имеющейся в генах информации. Методы генетики, например генетические алгоритмы, позволяют решать многие математические, физические и иные задачи с повышенной эффективностью.

Название азотистых оснований

Наименование основания	Русскоязычное обозначение	Англоязычное обозначение
Аденин	А	A
Гуанин	Г	G
Тимин	Т	T
Цитозин	Ц	C
Урацил	У	U

С позиций биохимии специфической особенностью живых организмов является наличие нуклеиновых кислот двух типов: дезоксирибонуклеиновой (ДНК) и рибонуклеиновой (РНК). Мономерами нуклеиновых кислот являются нуклеотиды. Последние содержат азотистые основания, приведенные в таблице.

ДНК являются материализацией генов и ответственны за программу зарождения, развития и смерти живого организма. А РНК ответственны за синтез белков, составляющих основу живого организма. В зависимости от своего предназначения РНК делится на информационные РНК, матричные мНРК (от matrix — матка), транспортные тРНК и рибосомные рРНК.

ДНК представляет собой молекулу, содержащую две последовательности — цепи, образующие спираль диаметром около 2 нм. В 1954 г. американец Дж. Уотсон и англичанин Ф. Крик предложили трехмерную модель спирали ДНК (рис. 1). Между спиралями, расположенными на расстоянии около 0,34 нм, наподобие винтовой лестницы, попарно располагаются азотистые основания, соединенные между собой водородными связями. Они удерживают спирали. Концы спиралей обозначают как 3′ и 5′ (цифры указывают на номера связей). При этом 3′ — конец одной спирали — соответствует 5′ концу другой, и наоборот. Поэтому говорят, что спирали комплементарны.

В этой модели каждая пара оснований содержит одно пуриновое и одно пиримидиновое основания: аденин с двумя водородными связями соединяется только с тимином, а цитозин посредством трех водородных связей связывается только с гуанином. Это схематично показано на рис. 1.

Молекула ДНК имеет большую длину. К примеру, в некоторых молекулах ДНК человека число пар оснований достигает $2 \cdot 10^9$. Отсюда ясны огромные трудности по разгадке кодов генов, которая упорно и успешно продолжается уже многие годы. Можно сказать, что природа подарила нам пример поразительно компактной записи всех особенностей живых организмов.

Запись информации о последовательности аминокислот является *генетическим кодом*. Последний в молекуле мРНК задается триплетами, получившими название *кодонов*. Комплементарные им триплеты в молекуле тРНК называют *антикодонами*. Триплеты задают 64 кодона, и все они выполняют определенные функции. Однако первые 20 кодов имеют особо важное и принципиальное значение [1].

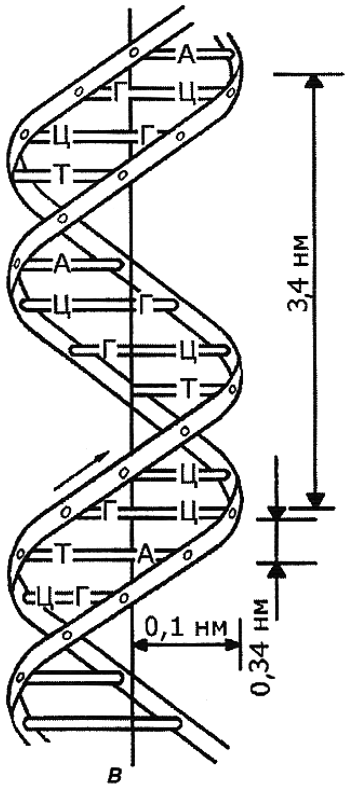


Рис. 1. Модель двойной спирали ДНК по Уотсону и Крику

Пакет Bioinformatics Toolbox 3.1 системы MATLAB R2008a

Одним из мощных инструментов решения задач биоинформатики является пакет расширения системы MATLAB Bioinformatics Toolbox. В новейшую версию 3.1 последней реализации системы MATLAB 2008a [3] входит множество функций, полезных для специалистов и ученых в этой области и существенно расширяющих возможности проведения научных исследований в биоинформатике и в обучении этой дисциплине. Функции данного пакета позволяют его пользователям создавать новые алгоритмы в области обработки биологической информации и генной инженерии. Пакет является серьезным развитием его более ранних версий, описанных в [4].

В основе биоинформатики лежат сравнительные исследования аминокислотных или геномных последовательностей, анализ и поиск сходства генных последовательностей. Именно это в первую очередь обеспечивает данный пакет.

Доступ к мировым информационным ресурсам

Для решения серьезных и актуальных в наше время задач биоинформатики необходим доступ к новейшим данным в той или иной области. Пакет расширения Bioinformatics Toolbox, наряду с небольшой своей демонстрационной базой данных по биологическим системам и генам, имеет возможность прямого доступа к мировым информационным ресурсам в области биологии и генетики, расположенным на сайте Национального центра биологической информации NCBI. Для этого достаточно ввести адрес этого центра в команду обращения встроенного в MATLAB Интернет-браузера:

> > web('http://www.ncbi.nlm.nih.gov/')

При наличии обычного подключения к Интернету это приведет к загрузке и появлению начального окна Интернет-страницы NCBI (рис. 2).

Работа с генными цепочками на примере вируса птичьего гриппа

Одна из важных возможностей пакета — работа с последовательностями генетического кода — включает в себя анализ, статистику, выравнивание и сравнение генетических цепочек, а также преобразование и композицию аминокислот. С помощью анализа цепочек можно найти информацию о нуклеотидной последовательности или последовательности аминокислот, используя реализованные в пакете специальные вычислительные методы.

Рассмотрим актуальную в наше время задачу изучения генов птичьего гриппа. Птичий грипп — это инфекционная болезнь птиц, вызываемая одним из штаммов вируса гриппа типа А. Из 15 подтипов вируса птичьего гриппа вирус H5N1 вызывает особенное беспокойство по ряду причин. Он очень быстро мутирует и, что подтверждено исследованиями, имеет склонность получать гены от вирусов, инфицирующих другие виды животных. По меньшей мере дважды подтверждена способность H5N1 вызывать тяжелую болезнь у человека. Чтобы вирус приобрел эту способность, достаточно замены всего одной аминокислоты в одном из вирусных белков.

С помощью пакета Bioinformatics Toolbox нетрудно получить данные о гене вируса птичьего гриппа из все-

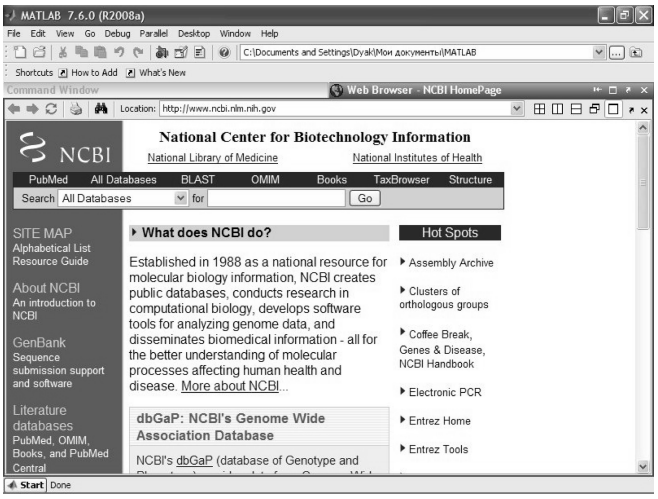


Рис. 2. Главная страница Интернет-сайта Национального центра биологической информации NCBI

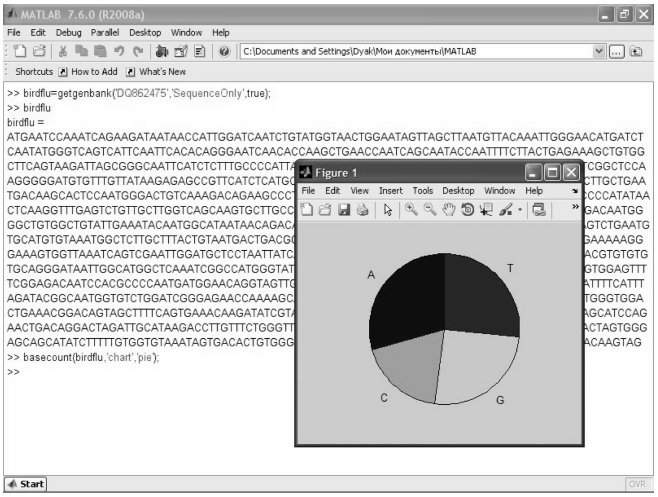


Рис. 3. Выявление состава генной цепочки птичьего гриппа

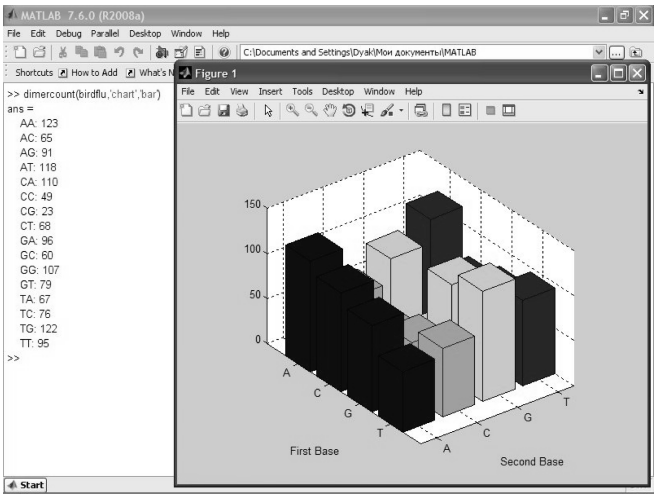


Рис. 5. Построение трехмерной гистограммы распределения компонент генной цепочки (гистограмма представлена в окне, расположенном в окне командного режима работы системы MATLAB)

мирной базы данных NCBI. На рис. 3 показано, как это делается для вируса с идентификационным номером DQ862475, который именуется как Influenza A virus (A / chicken / Navapur / Nandurbar/India/7972/ 2006(H5Nl). Справа показаны команды MATLAB для доступа к генетической цепочке выбранного вируса и ее относительный состав, представленный круговой диаграммой. Она строится функцией *basecount*(*birdflu*, *'chart'*, *'pie'*), где *birdflu* — имя массива с данными полученной цепочки.

Следующий пример показывает, как можно изобразить графики плотности азотистых оснований, а также графики разностей A-T и C-G (рис. 4, см. вторую сторону обложки). Имеется также возможность просмотреть, каким именно аминокислотам соответствуют те или иные триплеты. Для этого можно использовать функцию *aminolookup* (команда в окне MATLAB снизу).

Заинтересованный читатель может продолжить изучение данного вируса. Например, функция *dimercount* позволяет строить объемную гистограмму (рис. 5).

Преобразование нуклеотидной последовательности в аминокислотную происходит с помощью функции nt2aa:

```
> > BF2AaSeq = nt2aa(birdflu, 'geneticcode',
    'Vertebrate Mitohondrial');
```

Преобразование и композиция аминокислот

Применение функции *aacount* позволяет получить данные о наличии в рассматриваемой цепочке различных аминокислот и построить гистограмму по этим данным (рис. 6)

```
> > aacount(BF2AaSeq, 'char', 'bar')
ans = A: 19 R: 3 N: 30 D: 21 C: 18 Q: 11 E: 19 G: 44 H: 11
      I: 20 L: 19 K: 18 M: 22
      F: 18 P: 22 S: 54 T: 28
      W: 15 Y: 11 V: 31 Others: 16
```

Полученные детальные данные о составе генетических цепочек позволяют средствами пакета выполнять выравнивание и сравнение генетических цепочек.

Средства работы с микромассивами

Клетки живых организмов и растений представляют собой сложнейшие микроскопические образования, для которых в пакете Bioinformatics Toolbox используется термин *микромассивы*. Его не надо путать с определением размеров массива как числа его элементов. Типичными примерами микромассивов являются микрофотографии клеток живых организмов и растений.

Демонстрационные микромассивы определены в файлах с расширением .gpr, которые находятся в директории TOOLBOX/BIOINFO/DEMOS системы MATLAB. Пример загрузки, просмотра структуры и визуализации микромассива с данными о генах мыши показан на рис. 7 (см. вторую сторону обложки).

Пакет Bioinformatics Toolbox имеет ряд средств обработки микромассивов. Так, команда colormap hot обеспечивает улучшение цветовой гаммы изображений микромассивов. Команда colormapeditor выводит окно редактора цветовой карты микромассивов. Есть возможность сравнения микромассивов, например, для здоровых и больных организмов, а также возможность построения статистических диаграмм, облегчающих диагностику заболеваний. Возможно построение графиков профиля

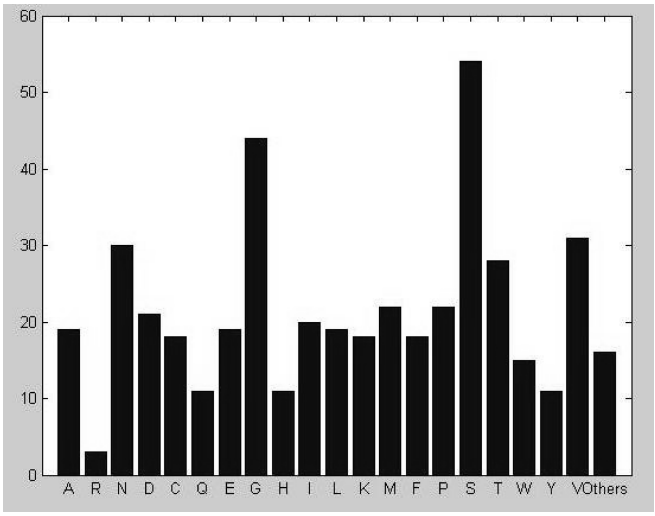


Рис. 6. Гистограмма присутствия азотистых оснований в цепочке BF2AaSeq

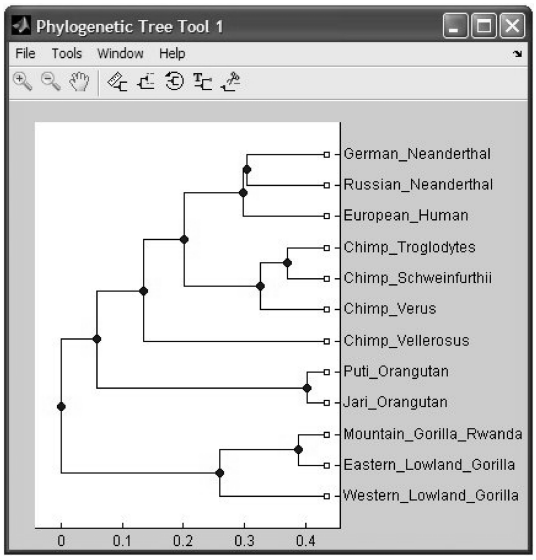


Рис. 8. Филогенетическое дерево приматов

микромассивов, а также построение карт кодонов, содержащих шесть фреймов.

Построение филогенетических деревьев

Многие биологические объекты имеют значительную степень связанности. Связь между ними удобно представлять филогенетическими деревьями. Пакет Bioinformatics Toolbox имеет мощные средства для построения таких деревьев. Но мы ограничимся простым примером загрузки последовательности объектов — предков человека (приматов), создания их генетического дерева и его построения (рис. 8):

```
> > load primatesdemodata
> > tree = seqlinkage(seqpdist(primates), 'single', primates);
> > view (tree)
```

Кластеризация и анализ основных составляющих генов

Скопления генов со схожими характеристиками называют *кластерами*. Пакет Bioinformatics Toolbox имеет

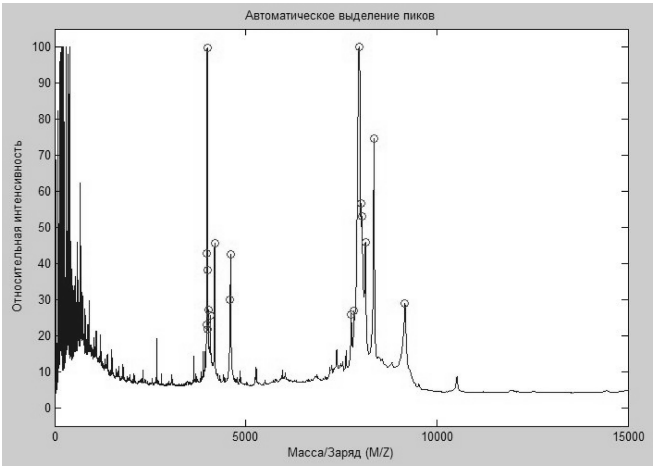


Рис. 11. Спектр с отмеченными пиками

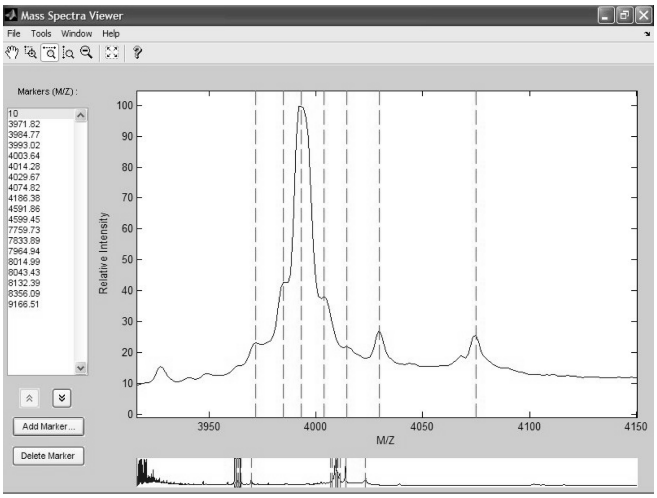


Рис 12. Окно вьювера спектра

различные средства для выявления кластеризации и построения. Их описание можно найти в [2]. Хотя это описание относится к более ранним версиям пакета, оно вполне применимо для новейшей версии V3.1. Средства пакета позволяют осуществлять кластеризацию, строить профили и деревья кластеров и осуществлять анализ основных составляющих генов с его визуализацией в окне Principal Component Analysis (PCA). Работа с этим простым окном очевидна.

Самоанализ кластеров с применением нейронных сетей

Пакет Bioinformatics Toolbox может применяться совместно с другим мощным пакетом расширения системы MATLAB — по нейронным сетям. Функция этого пакета *newsom* позволяет создать нейронную сеть типа SOM (Self Organizing Map), которую (после обучения по умолчанию) можно использовать для нахождения центров кластеров и соединения их отрезками прямых с использованием тех или иных критериев (рис. 9, см. вторую сторону обложки).

Визуализация молекул

Входящий в пакет Bioinformatics Toolbox инструмент визуализации Visualization Tools обеспечивает дополнительные возможности в наблюдении кластеров, применении графов и в пространственной визуализации молекул. В качестве примера на рис. 10 (см. вторую сторону обложки) показана визуализация молекулы аспирина в окне инструмента визуализации.

Средства визуализации масс-спектрометрического анализа

В биологии исключительно велика роль масс-спектроскопического анализа. Он, к примеру, позволяет определить наличие и концентрацию в исследуемом материале различных его составляющих. Массовые данные спектрометрии обычно сохраняются в текстовых файлах осциллографического формата CVS с двумя столбцами, хранящих данные о массе/нагрузке (M/Z) и значениях интенсивности, соответствующие этим M/Z отношениям. Для загрузки данных используют одну из следующих функций MATLAB ввода-вывода: `importdata`, `csvread`, или `textread`. Можно использовать также функцию `jcampread`, чтобы загрузить JCAMP-DX отформатированные файлы и `xlsread`, чтобы загрузить файлы электронных таблиц в формате рабочих книг Excel.

Рассмотрим пример загрузки файла mspec0l.csv с помощью функции importdata:

```
sample = importdata('mspec01.csv')
sample = data: [15154x2 double]
textdata: {'M/Z' 'Intensity'}
colheaders: {'M/Z' 'Intensity'}
```

Из полученного массива sample можно выделить подмассивы:

```
MZ = sample.data(:,1); Y = sample.data(:,2);
```

Теперь можно создать программу построения спектра с отметкой пиковых точек, за исключением низкочастотных и малых по высоте пиков (рис. 11):

```
slopeSign = diff(YS(:,1)) > 0;
slopeSignChange = diff(slopeSign) < 0;
h = find(slopeSignChange) + 1; %Нахождение всех пиков
h(MZ(h) < 1500) = []; %Удаление низкочастотных пиков
h(YS(h,1) < 20) = []; %Удаление малых по высоте пиков
plot(MZ, YS(:, 1), '-',MZ(h), YS(h, 1), 'ro')
axis([0 15000-5 105])
xlabel ('Масса/Заряд (M/Z)');ylabel('Относительная
интенсивность') title ('Автоматическое выделение пиков')
```

Пакет расширения имеет вьювер спектра. Он открывается (для нашего примера) командой *msviewer* (*MZ, YS, 'MARKERS', MZ*(*h*)) и открывает окно, показанное на рис. 12. Нижнее подокно дает обзор спектра с маркерами, указывающими перемещаемую область обзора. Верхнее подокно дает участок спектра. Пики выделяются вертикальными штриховыми линиями — маркерами (их список дан слева).

В окне вьювера спектра можно просматривать сразу несколько спектров. Пакет обеспечивает также представление трехмерных спектров и построение спектрограмм, подобных спектрограммам быстрого оконного спектрального анализа. Для обработки спектров могут использоваться генетические алгоритмы (реализуются пакетом расширения Genetic Algorithm and Direct Search Toolbox^{TM}), а также средства параллельных вычислений. Есть основания полагать, что мощные средства масс-спектроскопического анализа и его визуализации могут

использоваться для решения задач спектрального анализа не только в области биоинформатики, но и во многих других областях науки и техники.

Заключение

Новейшая реализация пакета биоинформатики Bioinformatics Toolbox 3.1 системы MATLAB R2008a достойно продолжает развитие ее пакета по биоинформатике, который предоставляет пользователю превосходный набор инструментов для работ в области биологии, биоинформатики, математики, физики и во многих областях науки и техники, использующих методы биоинформатики. Новый пакет обеспечивает работу с огромной информацией о генах и генетике, размещенной в Интернете. Трудно переоценить роль пакета в системе образования, где он позволяет на современном уровне изучать самые современные методы биоинформатики и генной инженерии.

Список литературы

1. **Биология.** Специальный курс / Под ред. А. Ф. Никитина. СПб.: СпецЛит, 2005. 480 с.
2. **Афанасьева Г.** Биоинформатика: виртуальный эксперимент в шаге от реальности // Наука и жизнь. 2004. № 11. С. 20.
3. **Дьяконов В. П.** MATLAB R2006/2007.2008 + Simulink 5/6/7. Основы применения. 2-е изд., переработ. доп. Сер. "Библиотека профессионала". М.: Солон-ПРЕСС, 2008. 800 с.
4. **Дьяконов В. П., Круглов В. В.** MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. Сер. "Библиотека профессионала". М.: Солон-ПРЕСС, 2006. 456 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК 004:519.2:37

Е. Ю. Карданова, канд. физ.-мат. наук, доц.,
В. Б. Карпинский, преподаватель,
Новгородский государственный
университет имени Ярослава Мудрого,
г. Великий Новгород,
e-mail: e_kardanova@mail.ru

Использование эксперимента на модели Раша для выявления недостоверных результатов педагогического тестирования

При анализе результатов массового тестирования использование традиционных статистических критериев согласия оказывается недостаточно эффективным для выявления недостоверных результатов тестирования. Приведены результаты исследования, включающего ряд вычислительных экспериментов на модели Раша, которое позволило не только выяснить причины возникновения этой проблемы, но и разработать технологию обработки информации, повышающую эффективность обнаружения недостоверных результатов тестирования.

Ключевые слова: *тестирование, выявление искажений, статистика согласия, нормированное уклонение, критическое значение, модель Раша, вычислительный эксперимент.*

Введение

Известно [1—3], что использование математических моделей Г. Раша позволяет подойти к оцениванию знаний как к процессу объективного измерения, что подразумевает получение инвариантных относительно друг друга оценок мер испытуемых и параметров заданий теста на единой метрической шкале и с указанием точности оценивания.

Рассмотрим модель Раша с произвольными промежуточными категориями выполнения заданий [3], наиболее часто используемую в педагогическом тестировании. Согласно этой модели вероятность P_{nik} того, что испытуемый n с уровнем подготовленности θ_n получит k баллов за выполнение задания i (т. е. выполнит k шагов в этом задании), $k = 0, 1, ..., m_i$, определяется формулой

$$P_{nik} = \frac{\exp\left(k\theta_n - \sum\limits_{j=0}^{k} \delta_{ij}\right)}{\sum\limits_{l=0}^{m_i} \exp\left(l\theta_n - \sum\limits_{j=0}^{l} \delta_{ij}\right)}, \qquad (1)$$

где σ_{ij} — трудность выполнения шага j задания i, $\delta_{i0} = 0$, m_i — максимальный балл в задании i. Модель (1) называется политомической моделью Раша. Величины θ_n, $n = 1, ..., N$, и δ_{ij}, $i = 1, ..., I$; $j = 1, ..., m_i$, являются параметрами модели и могут быть оценены обычными методами.

Помимо педагогического тестирования технология обработки данных, основанная на моделях Раша, применима для решения многих задач в других областях (управлении, медицине, экономике и т. д.), где требуется по матрице некоторых показателей оценить латентные параметры объектов любой природы. Например, в работе [4] технология обработки информации, использующая модель Раша, применяется для обработки данных многокритериального мониторинга в региональном управлении качеством образования.

Возможности модели Раша не ограничиваются оцениванием латентных параметров. В настоящее время актуальной задачей, помимо прочего, является обнаружение в измерении испытуемых искажений, обусловленных нарушениями процедуры тестирования (например, списыванием, подлогом и т. д.) и приводящих к недостоверным результатам при массовых тестированиях. Некоторые методы обнаружения недостоверных результа-

тов тестирования представлены в работе [2]. Однако при массовых тестированиях необходимы статистические методы, позволяющие оперативно обрабатывать большие потоки информации и надежно выявлять недостоверные результаты.

Нормированное уклонение

$$y_{ni} = \frac{x_{ni} - M(x_{ni})}{\sqrt{D(x_{ni})}}, \qquad (2)$$

где x_{ni} — балл, полученный испытуемым n за задание i, $M(x_{ni})$ и $D(x_{ni})$ —соответственно математическое ожидание и дисперсия этой случайной величины согласно модели (вычисленные с учетом (1)), является базовой статистикой, так как на ее основе строится целый ряд статистических критериев, традиционно используемых для обнаружения искажений в измерении испытуемых [5, 6]. В качестве критического значения, как правило, используется соответствующий квантиль теоретического распределения статистики. При этом основополагающим является предположение о характере распределения нормированных уклонений (2). Например, на основе предположения о стандартизованном нормальном $N(0; 1)$ распределении статистики (2) общая статистика согласия

$$U_n = \frac{1}{I} \sum_{i=1}^{I} y_{ni}^2, \qquad (3)$$

где I — число заданий в тесте, имеет распределение Фишера— Снедекора $F(I, \infty)$. Его квантиль на уровне значимости $\alpha = 0{,}05$ составляет 1,35, что и является теоретическим критическим значением.

В работе [7] исследована эффективность статистических критериев согласия для обнаружения недостоверных результатов тестирования; она была признана недостаточно высокой, и был предложен комплексный критерий с более высокой эффективностью. В работах [8, 9] было предложено выбирать его критическое значение (а также критические значения входящих в его состав статистик согласия), минимизируя суммарные потери от ошибок первого и второго рода на модельных данных с достоверно известными искажениями. Однако причины того, что стандартные, имеющие теоретическое обоснование статистические критерии согласия недостаточно эффективны для задачи выявления искажений, требовали более внимательного анализа.

В настоящей статье представлены результаты исследования, позволившего не только выяснить причины возникновения указанной проблемы, но и разработать технологию обработки данных, повышающую эффективность выявления искажений в результатах тестирования и обнаружения недостоверных результатов.

Особенности распределения нормированных уклонений

Было высказано предположение, что основной причиной недостаточной эффективности статистик согласия является отличие эмпирического распределения базовой статистики (2) от $N(0; 1)$. В работе [10] проведено исследование свойств распределения статистики (2) на специально сконструированных модельных данных и отмечены более "толстые" по сравнению со стандартизованным нормальным распределением "хвосты" распределения. Однако делается вывод, что в большинстве случаев его можно приближенно считать стандартизованным нормальным $N(0; 1)$.

С учетом ограниченности исследования, проведенного в [10], было решено выполнить более широкое исследование случайной величины (2) с использованием результатов ЕГЭ за прошлые годы по нескольким предметам, выбираемых так, чтобы отразить различные аспекты, возможно, влияющие на вид эмпирического распределения (число испытуемых и задании, наличие или отсутствие смещения совокупности испытуемых относительно совокупности заданий, различный разброс значений параметров и т. д.).

Для вычислительного эксперимента были использованы серии модельных матриц, генерируемых специально разработанной программой в соответствии с моделью Раша (т. е. заведомо без искажений). Параметры модели (уровни подготовленности испытуемых θ_n и уровни трудности заданий δ_{ij}) либо заимствовались из указанных реальных данных, либо формировались с заданными параметрами распределения. Было экспериментально проверено, что статистически достаточно генерировать по 10 матриц каждого типа.

В табл. 1 представлены параметры эмпирического распределения нормированных уклонений (2), усредненные по 10 матрицам. Для демонстрации того, что прецизионность серии опытов достаточно высока, указаны соответствующие среднеквадратические отклонения (СКО).

Из табл. 1 видно, что в части асимметрии и, особенно, эксцесса имеются существенные отличия эмпирических распределений от теоретического, стандартизованного нормального распределения $N(0; 1)$.

В табл. 2 в качестве характеристики формы эмпирического распределения нормированных уклонений приведены усредненные по 10 матрицам частоты отклонения их значений от нуля более чем на заданную величину. В последней строке таблицы приведены соответствующие частоты для стандартизованного нормального распределения.

Таблица 1

Параметры эмпирического распределения нормированных уклонений по всем предметам

Предмет-прототип	Среднее		Стандартное отклонение		Асимметрия		Эксцесс	
	Среднее	СКО	Среднее	СКО	Среднее	СКО	Среднее	СКО
География	0,000	0,001	1,001	0,005	−0,391	0,124	5,197	1,631
История	−0,001	0,001	1,001	0,001	−0,076	0,047	2,427	0,312
Литература	−0,001	0,003	0,999	0,011	−1,758	0,385	18,692	7,244
Математика	−0,002	0,003	1,013	0,011	−0,007	1,318	101,294	40,910
Обществоведение	−0,001	0,001	1,002	0,003	−1,137	0,093	6,525	1,791
Среднее	−0,001		1,003		−0,710		15,153	
СКО	0,001		0,006		0,863		32,828	

Таблица 2

Частоты отклонения нормированных уклонений от нуля более чем на указанную величину

Предмет-прототип	<−4	<−3	<−2	<−1	>1	>2	>3	>4
География	0,0023	0,0067	0,0257	0,1368	0,1125	0,0175	0,0041	0,0013
История	0,0014	0,0048	0,0211	0,1356	0,1300	0,0189	0,0040	0,0013
Литература	0,0050	0,0104	0,0286	0,1159	0,1016	0,0274	0,0155	0,0031
Математика	0,0045	0,0088	0,0209	0,0729	0,0740	0,0240	0,0128	0,0015
Обществоведение	0,0033	0,0092	0,0335	0,1528	0,0907	0,0108	0,0021	0,0007
Среднее	0,0033	0,0080	0,0260	0,1228	0,1018	0,0197	0,0077	0,0016
Стандартное отклонение	0,0015	0,0022	0,0053	0,0308	0,0212	0,0064	0,0060	0,0009
Теоретическое	0,00004	0,0016	0,0255	0,1710	0,1710	0,0255	0,0016	0,00004

Из табл. 2 следует вывод о наличии закономерных отклонений эмпирических распределений от теоретического распределения. Общий характер этих отклонений можно выразить словами "толстые хвосты".

На рис. 1 представлена гистограмма распределения статистики (2) для наиболее типичного случая (модель на основе предмета-прототипа "география"), где данные сгруппированы в оптимальное число групп K ($K = 1{,}72\ N^{1/3}$, где N — объем выборки [11]; для $N = 32750$ $K = 55$). Мелкими точками показано рассеяние значений статистики, а тонкой линией — график функции плотности распределения $N(0; 1)$. На рис. 1, как и на всех последующих рисунках, по оси абсцисс откладываются значения статистик, а по оси ординат — их частоты (для лучшего восприятия рисунка ось ординат смещена относительно нуля оси абсцисс).

Очевидны два вида отличий эмпирического распределения от стандартизованного нормального распределения. Во-первых, это "щель" в вершине гистограммы, т. е. ярко выраженная бимодальность распределения. Во-вторых, слишком толстые "хвосты". Для задачи выявления искажений вторая особенность эмпирического распределения важнее, так как интересующие нас квантили распределения располагаются именно на его "хвосте" (процент искажений обычно невелик).

Интересно, что χ^2-критерий даже такое распределение, как показано на рис. 1, при более грубом подходе признает соответствующим стандартизованному нормальному распределению. Однако при указанном выше оптимальном числе групп 55 гипотеза о соответствии эмпирического распределения исследуемой статистики теоретическому распределению $N(0; 1)$ отвергается (значение χ^2-критерия равно 327,06; критическое значение при уровне значимости 0,05 равно 73,38 [11]).

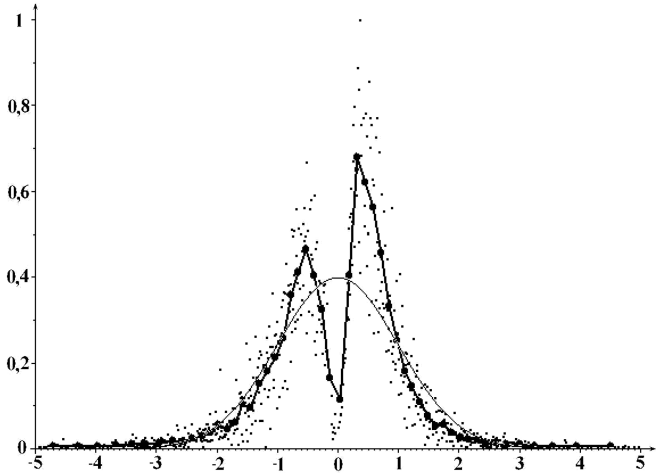


Рис. 1. Гистограмма распределения нормированных уклонений (толстая линия и точки) в сравнении с графиком функции плотности стандартизованного нормального распределения (тонкая линия)

Подобная ситуация наблюдается во всех остальных аналогичных экспериментах: при более грубом подходе распределение можно считать нормальным, а при менее грубом — нельзя.

Несколько серий модельных экспериментов показали, что наличие двух мод, соотношение их высот, размер "щели" между ними и другие особенности эмпирического распределения статистики (2) зависят от распределения параметров модели. Это позволило сформулировать рекомендации по составлению тестов, при соблюдении которых распределение нормированных уклонений наиболее близко к $N(0; 1)$. Чтобы тест был "хорошим" в указанном смысле, желательно, чтобы трудности заданий и уровни подготовленности испытуемых:

- были хорошо взаимно центрированы на шкале логитов;
- имели перекрывающие друг друга диапазоны варьирования;
- имели радиус этих диапазонов, близкий к трем логитам (логит — это единица измерения подготовленности испытуемых и трудности заданий на единой метрической шкале, см. [1]).

Следует отметить, что эти же требования предъявляются к тесту для обеспечения минимальной ошибки измерения испытуемых [3] и более или менее соответствуют выводам работы [10].

При теоретическом анализе нетрудно показать, что полимодальность имманентна распределению нормированных уклонений, хотя и должна нивелироваться, если и уровни подготовленности испытуемых, и уровни трудности заданий варьируются не менее чем от −3 до +3 логитов. Рис. 2 наглядно демонстрирует это. Были сгенерированы модельные матрицы без искажений с совпадающими диапазонами варьирования уровней подготовленности испытуемых и трудности заданий разного радиуса. Видно, что при радиусе 1 логит бимодальность ярко выражена, при радиусе 2 логита моды распределения несколько сближаются, а при радиусе 3 логита — почти сливаются.

Для исследования того, как изменяется характер распределения нормированных уклонений при наличии искажений в матрице ответов, генерировались серии матриц с такими же параметрами модели Раша, но с добавлением известного числа профилей испытуемых, содержащих заведомые искажения типа списывания/подлога. Было выявлено, что влияние искажений закономерно сказывается в виде усиления ранее обнаруженных дефектов распределения нормированных уклонений.

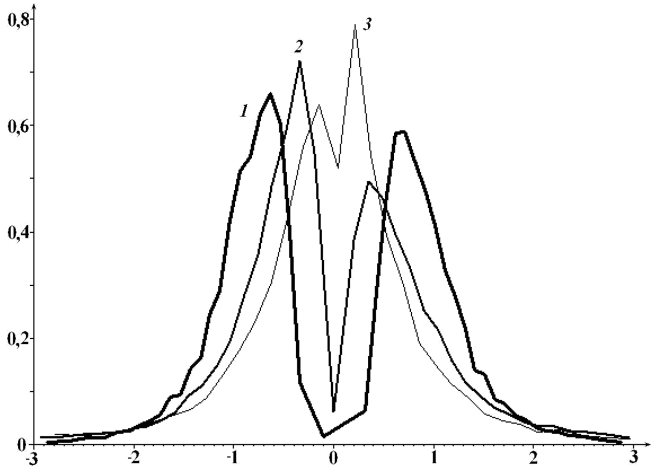


Рис. 2. Гистограммы распределения нормированных уклонений на моделях с диапазоном варьирования параметров:
1 — от −1 до +1; *2* — от −2 до +2; *3* — от −3 до +3

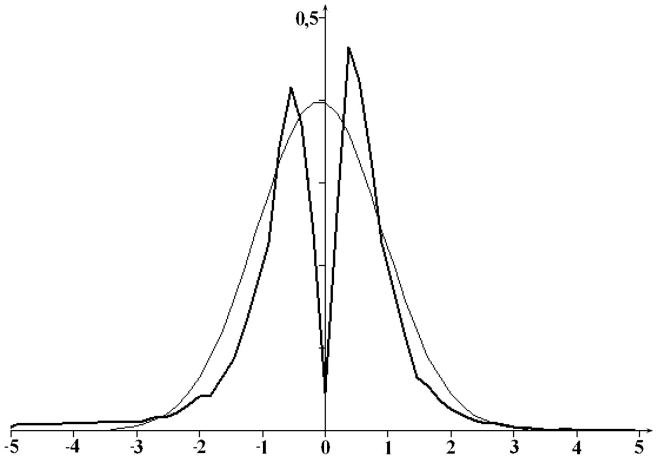


Рис. 3. Гистограммы распределения нормированных уклонений, преобразованных по формуле (4) (толстая линия), и график функции плотности стандартизованного нормального распределения (тонкая линия)

"Щель" между модами становится больше, "хвосты" распределения — толще. Из этого можно сделать два вывода. Во-первых, именно при наличии в результатах тестирования искажений предположения, основанные на характере теоретического распределения используемых статистик, могут оказаться неверны. Во-вторых, по "щели" и "хвостам" распределения нормированных уклонений на реальных данных тестирования можно высказать предварительное предположение о наличии (и, возможно, даже количестве) искажений в них.

Наличие большого числа политомических заданий (с максимальным баллом больше единицы) может несколько изменить характер распределения нормированных уклонений (сделать выраженными более чем две моды или, напротив, размыть и завуалировать полимодальность), но не может сделать это распределение нормальным. Приближенный, и даже зачастую грубо приближенный, характер замены распределения нормированных уклонений на стандартизованное нормальное распределение — неустранимая особенность этой статистики.

Аппроксимация эмпирического распределения

Традиционным методом решения проблемы недостаточного соответствия эмпирического распределения теоретическому является аппроксимация эмпирического распределения. В настоящем исследовании был применен метод аппроксимации S_U-кривыми Джонсона [12], которые получаются преобразованием выборки с помощью функции, зависящей от четырех параметров. По этим параметрам оптимизируется близость распределения преобразованной выборки к распределению $N(0; 1)$. Преобразование Джонсона обычно используется только для унимодальных распределений. Поэтому было выполнено два типа преобразования:

- общее для всей совокупности нормированных уклонений вне зависимости от бимодальности их распределения;
- отдельно для каждой унимодальной подвыборки с последующим объединением.

В первом случае для рассматриваемого примера с предметом-прототипом "география" хорошее приближение к стандартизованному нормальному распределению (рис. 3) дает преобразование по формуле

$$y'_{ni} = 0{,}36 + 1{,}76\text{arcsh}((y_{ni} - 0{,}35)/1{,}45). \qquad (4)$$

После преобразования получаем среднее −0,0008, стандартное отклонение 1,0018, асимметрию 0,0039 и эксцесс −0,0014 (см. табл. 1 для сравнения с соответствующими значениями до преобразования). Новое распределение признается χ^2-критерием, стандартизованным нормальным на отрезке [−5; 5] при оптимальной группировке в 55 карманов (значение критерия 25,24 при критическом значении 73,78). Видно, что "щель" (бимодальность) хотя и не исчезла, не мешает χ^2-критерию признавать полученное распределение нормальным. Однако на отрезке [−6; 6] гипотеза о нормальности распределения отвергается — сказывается толстый левый "хвост" распределения.

Применение аппроксимации Джонсона с разделением мод для того же примера (рис. 4) дает следующие формулы преобразования:

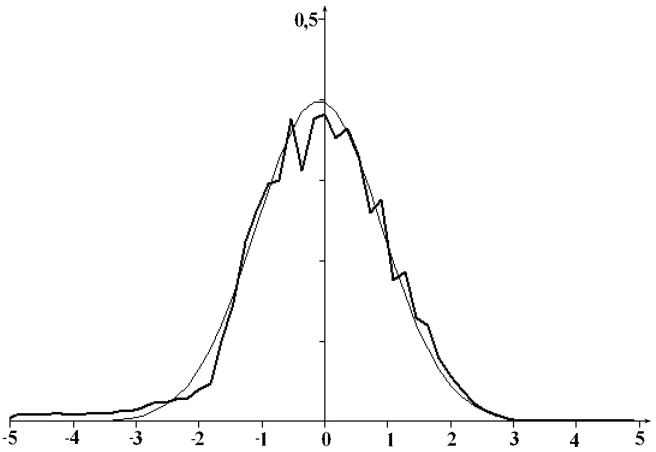


Рис. 4. Гистограммы распределения нормированных уклонений, преобразованных по формулам (5) (толстая линия), и график функции плотности стандартизованного нормального распределения (тонкая линия)

$$y'_{ni} = -0{,}7377 + 1{,}07\text{arcsh}((y_{ni} - 0{,}35)/0{,}3) \text{ при } x_{ni} > 0;$$
$$y'_{ni} = 4{,}1053 + 1{,}53\text{arcsh}((y_{ni} - 0{,}02)/0{,}1) \text{ при } x_{ni} = 0. \quad (5)$$

Моменты распределения также близки к моментам $N(0; 1)$: среднее 0,0000, стандартное отклонение 1,0022, асимметрия −0,0113 и эксцесс 0,0236. Нормальность этого распределения принимается и отвергается χ^2-критерием в тех же случаях, что и у получаемого преобразованием (4).

Распределение статистик согласия и их критические значения

Рассмотрим, какое влияние оказывают особенности распределения базовой статистики (2) на распределения статистик согласия. С этой целью для всех серий модельных матриц вычислялись также и значения различных статистик согласия для каждого испытуемого. Профиль испытуемого считается содержащим искажения, если значения статистик согласия превосходят некоторое (свое для каждой статистики) критическое значение. Из этих опытов следовало, что теоретические критические значения не дают достаточно высокой эффективности выявления недостоверности результатов тестирования. И причиной этого являются именно толстые "хвосты" распределения нормированных уклонений.

Рассмотрим более подробно одну из статистик — общую статистику согласия U_n (3). Интересно, что преобразование (5), дающее распределение нормированных уклонений, которое лучше соответствует теоретическому распределению (рис. 4), при переходе к статистике согласия оказывается хуже, чем преобразование (4), сохраняющее "щель" и бимодальность. В частности, при оптимальном числе групп (15 для объема выборки 655) χ^2-критерий имеет для преобразованной с разделением мод (5) статистики U_n значение 37,96, а при преобразовании без разделения мод (4) — значение 24,68 (при критическом значении 25,01 на уровне значимости $\alpha = 0{,}05$). Таким образом, в первом случае соответствие распределения теоретическому распределению $F(I, \infty)$ отвергается, а во втором — принимается. Как мы и ожидали, толстые "хвосты" существенно влияют на эффективность статистик согласия, а "щель" (бимодальность) — несущественно.

Эффективность статистик согласия при моделировании без искажений может быть оценена по числу ложных срабатываний критерия (по частоте ошибки второго рода). Если статистика U_n (3) имеет теоретическое распределение $F(I, \infty)$, то 5 % ложных срабатываний дает использование критического значения 1,35. Для статистики, преобразованной по формулам (5), соответствующий квантиль составляет 1,75, для исходной, не преобразованной, — 1,6. Сказывается толстый "хвост" распределения.

Все статистики согласия были исследованы и на модельных данных, содержащих заведомо искаженные профили. В этом случае в качестве показателя эффективности используется также частота пропуска искаженных профилей (частота ошибки первого рода). Результаты этой части исследования согласуются с представленными в настоящей статье выводами, хотя и заслуживают более подробного изложения.

Таким образом, несмотря на улучшение в результате аппроксимации соответствия распределения базовой статистики (2) теоретическому распределению $N(0; 1)$, важная для выявления искажений в результатах тестирования особенность этой статистики (толстый "хвост") осталась неустраненной. Этим подтверждается сделанный ранее вывод о ее неустранимости. И, следовательно, предлагаемый авторами метод выбора критического значения статистик согласия путем эмпирической оптимизации на специально сконструированных модельных данных остается на данный момент единственным, позволяющим достичь требуемого уровня эффективности методом.

Заключение

В результате как теоретического анализа, так и серии модельных экспериментов выяснена причина недостаточной эффективности статистических критериев, традиционно используемых для выявления недостоверных результатов тестирования. Эта причина коренится в неустранимом отличии распределения нормированных уклонений от стандартизованного нормального распределения, предположение о котором лежит в основе определения теоретического характера распределения всех статистик, строящихся на основе нормированных уклонений (2). Это отличие невелико, и при более грубом подходе распределение нормированных уклонений может считаться стандартизованным нормальным. Но определение критического значения статистического критерия для эффективного выявления искажений в результатах тестирования требует более тонкого, детального подхода, и в этом случае выводы, основанные на теоретическом характере распределения, становятся недостоверными.

Однако отработанная в ходе исследования технология проведения вычислительных экспериментов на модели Раша открывает новый путь повышения эффективности анализа результатов тестирования. На первом этапе строится модель без искажений по параметрам реальных данных. Сравниваются характеристики распределения нормированных уклонений (2) на модели и в реальных данных. Из степени их расхождения можно сделать предварительный прогноз о числе искаженных профилей в реальных данных. На втором этапе строится модель с таким числом искажений. По ней определяются оптимальные критические значения для всех используемых статистических критериев. На третьем этапе, также с оптимизацией на модельных данных, все показавшие хорошую эффективность статистические критерии объединяются в единый комплексный критерий. При необходимости вся процедура может быть повторена с изменением параметров модели. Важно подчеркнуть, что все шаги этого процесса автоматизированы путем разработки специального программного обеспечения.

Таким образом, не только проведено исследование причин недостаточного для наших целей соответствия эмпирического распределения статистик согласия их теоретическому распределению, но и разработана технология использования вычислительного эксперимента на модели Раша для повышения эффективности обработки данных массового тестирования в целях выявления в них искажений, т. е. недостоверных результатов тестирования.

Список литературы

1. **Нейман Ю. М., Хлебников В. А.** Введение в теорию моделирования и параметризации педагогических тестов. М.: Прометей, 2000. 169 с.

2. **Карданова Е. Ю., Карпинский В. Б.** Специальные методы анализа результатов тестирования, основанные на свойстве объективности моделей Раша // Информационные технологии. 2008. № 4 (140). С. 72—80.
3. **Карданова Е. Ю.** Моделирование и параметризация тестов: основы теории и приложения. М.: Федеральный центр тестирования, 2008. 304 с.
4. **Карданова Е. Ю., Карпинский В. Б.** Технология обработки информации в многокритериальном мониторинге на основе политомической модели Раша // Системы управления и информационные технологии. 2007. № 3. 1 (29). С. 149—154.
5. **Smith R. M.** Person and Item Analysis. Chicago: Mesa Press, 1992.
6. **Wright B. D., Masters G. N.** Rating Scale Analysis. Rasch Measurement. Chicago: Mesa Press, 1979. 206 p.
7. **Карпинский В. Б.** Исследование эффективности общих статистик согласия для обнаружения искажений при массовом тестировании // Вопросы тестирования в образовании. 2006. № 1 (17). С. 7—14.
8. **Карданова Е. Ю., Карпинский В. Б.** О возможностях обнаружения искажений при массовых тестированиях // Моделирование и параметризация педагогических тестов. Матер. Международной конференции, Минск, Беларусь. 2007. С. 30—36.
9. **Карданова Е. Ю., Карпинский В. Б.** Обнаружение искажений при тестировании с использованием математической модели Г. Раша // Обозрение прикладной и промышленной математики. 2007. Т. 14. Вып. 4. С. 716—717.
10. **Smith R. M.** The Distributional Properties of Rasch Standardized Residuals // Educational and Psychological Measurement. 1988. V. 48. P. 657—667.
11. **Вероятностные** разделы математики / Под ред. Ю. Д. Максимова. СПб.: Иван Федоров, 2001. 588 с.
12. **Кобзарь А. И.** Прикладная математическая статистика М.: Физматлит, 2006. 814 с.

УДК 004.588 + 004.384 + 004.415.2

В. Е. Зюбин, канд. техн. наук, ст. научн. сотр.,
Институт автоматики и электрометрии СО РАН,
г. Новосибирск,
Новосибирский государственный университет,
e-mail: zyubin@iae.nsk.su

Использование виртуальных объектов для обучения программированию информационно-управляющих систем

Представлен метод создания виртуальных лабораторных стендов для обучения программированию управляющих алгоритмов в области промышленной автоматизации. В качестве языка программирования использован язык Рефлекс (также известный как "Си с процессами"). Имитаторы контролируемых технологических объектов создаются в среде LabVIEW. Управляющие алгоритмы, описанные на языке Рефлекс, преобразуются к формату, позволяющему интегрировать алгоритм в среду LabVIEW через механизм Formula Node. Предлагаемый подход направлен на повышение эффективности процесса обучения, поскольку позволяет студентам исключить из рассмотрения рутинные вопросы физического моделирования и сконцентрироваться на концептуальном уровне программирования управляющих алгоритмов.

Ключевые слова: *обучение программированию, программные имитаторы, управляющие алгоритмы, язык Рефлекс.*

Введение

Качество образования, получаемого студентом, определяется не только уровнем теоретической подготовки, но и умением использовать полученные знания на практике. Практические навыки студент приобретает через специально предусматриваемые в учебном плане лабораторные работы и семинарские занятия. Достигаемые при этом эффекты — снижение "порога вхождения" в изучаемую область, сокращение времени на освоение материала, повышение уровня понимания теоретических положений. Для повышения уровня практической подготовки научных и технических специалистов в последние годы все более широкое применение находят информационные технологии, в частности, при организации лабораторных практикумов [1].

При изучении языков программирования, ориентированных на создание управляющих алгоритмов, требуется, чтобы лекционные курсы по информационно-управляющим системам (ИУС) были подкреплены лабораторно-практическими занятиями по решению типовых задач из области промышленной автоматизации. Однако лабораторные стенды, предполагающие физическое моделирование технологического процесса даже в упрощенном виде, означают серьезные финансовые затраты на их создание и поддержку в работоспособном состоянии. При таком подходе необходимо не только разработать и реализовать модель технологического процесса, но также оснастить ее датчиками, исполнительными органами и обеспечить сопряжение с компьютером через модули ввода-вывода. Стенды громоздки, занимают много места, а ошибки, допускаемые студентами при работе с ними, зачастую приводят к выходу стенда из строя.

В этих условиях наиболее перспективный способ организации практических занятий для студентов по курсам промышленной и физико-технической автоматизации должен быть основан на использовании программных имитаторов — виртуальных объектах управления (ВОУ). По сравнению с физическими моделями ВОУ имеют ряд очевидных преимуществ, дающих существенное сокращение материальных и временных затрат на создание, тиражирование и сопровождение лабораторных стендов.

В статье излагается методика создания лабораторных стендов на базе ВОУ, ориентированных на изучение основ программирования ИУС. Структурно статья представлена тремя разделами. В первом разделе с учетом особенностей изучения основ программирования ИУС формулируются требования к лабораторным стендам. Во втором разделе проводится анализ существующих средств моделирования и обсуждаются использованные программные и технологические средства создания ла-

бораторного стенда. Третий раздел посвящен вопросам реализации тестового лабораторного стенда.

Концепция виртуального лабораторного практикума по программированию информационно-управляющих систем

В качестве базовых требований к лабораторному стенду были выдвинуты следующие требования, обеспечивающие разумный баланс между наглядностью создаваемых имитаторов робототехнических комплексов, развитостью средств программирования алгоритмов управления и трудоемкостью реализации.

Требования, возникающие на этапе создания нового стенда:

- возможность создания анимированных 2D-объектов;
- допустимость использования при моделировании объекта автоматизации нескольких слоев с изображениями;
- наличие поведения у ВОУ, т. е. динамическая реакция ВОУ на внешние воздействия;
- возможность подключения стороннего алгоритмического блока, создаваемого студентом;
- стандартный интерфейс подключения стороннего алгоритмического блока, исключающий коррекцию ВОУ, а равно связей между ВОУ и сторонним алгоритмическим блоком в случае изменений алгоритма, создаваемого студентом.

Требования к этапу практической работы студента со стендом:

- наличие ручного и автоматического режима управления ВОУ;
- использование языка программирования, ориентированного на управляющие алгоритмы, для создания алгоритма управления ВОУ;
- наличие диагностической и отладочной информации.

Базовые программные и технологические средства

Чрезвычайно привлекательная идея использовать концепцию ВОУ для создания лабораторного практикума осложняется отсутствием программных средств, ориентированных на имитационное моделирование объектов автоматизации. Схожая задача создания операторских тренажеров решается на практике с серьезными трудозатратами: либо с использованием SCADA-пакетов, например InTouch (Wonderware), Shadow Plant (Honeywell), либо, чаще всего, вообще без использования специализированных пакетов, на языках Си/Си++ [2—5], явно не предназначенных для решения таких задач. При таком подходе цена тренажера может достигать 1 млн долл. [4]. Большинство широко известных языков имитации, таких как ARENA, Extend, WITNESS, QUEST, Enterprise Dynamics, Any-Logic и др., не имеют простых и мощных механизмов включения в модель правил и алгоритмов принятия решений [5], что не позволяет создавать на них "поведенческие" модели.

В связи с этим вполне определенный интерес представляют пакеты прикладных программ технических вычислений и системы автоматизации научных исследований типа MatLab и LabVIEW, широко используемые, в частности, как средства имитационного моделирования [6].

В результате анализа возможных претендентов на роль базовой среды программирования выбор был сделан в пользу пакета LabVIEW [7]. Хотя LabVIEW позиционируется как средство разработки программно-аппаратных комплексов для тестирования, измерения, ввода данных, анализа и управления внешним оборудованием, с точки зрения решаемой задачи пакет имеет целый ряд привлекательных свойств. Интерфейс пользователя позволяет не только отображать результаты в виде графиков и целого спектра графических элементов, но также работать с изображениями в tiff и bmp форматах. Специальный механизм (блок-диаграмм) дает возможность программировать функции управления объектами графического интерфейса. Вполне очевидный интерес представляет и возможность оформить создаваемые имитационные модели в виде автономных модулей (.EXE) и в виде совместно используемых динамических библиотек (.DLL), предоставляемая в LabVIEW Pro, поскольку эта возможность позволяет тиражировать создаваемые стенды и исполнять их автономно от среды разработки. Выбор обусловлен также популярностью LabVIEW и широким использованием в обучающем процессе, в частности, для создания лабораторных стендов. И хотя такое использование затрагивает в основном аналоговые случаи (электротехника, оптика, термодинамика и т. п. [8]), опыт показывает, что средствами LabVIEW можно создавать и несложные поведенческие алгоритмы, в частности, дискретные и событийно-управляемые (см., например, [9]).

В качестве языка программирования алгоритмов управления был выбран язык Рефлекс [10]. Язык ориентирован на программирование управляющих алгоритмов в промышленной автоматизации и робототехнике: для систем, предполагающих активное взаимодействие с внешней средой, технологическим оборудованием, физическими процессами через датчики и органы управления. Язык по синтаксису очень похож на язык Си (язык известен также под именем "Си с процессами"), что обеспечивает простоту его изучения большинством практикующих программистов. Язык имеет англоязычный и русскоязычный синтаксис, а также допускает идентификаторы на русском языке, и это делает его крайне привлекательным для русскоязычных пользователей. В отличие от языка Си, где программы строятся как иерархия функций, базовое понятие языка Рефлекс — процесс. Программа на языке Рефлекс — это множество параллельно исполняемых процессов, которые могут запускать друг друга, останавливать и контролировать текущее состояние.

Для интеграции алгоритмов, создаваемых на языке Рефлекс, в среду LabVIEW был использован механизм Formula Node. Этот механизм не позволяет исключить использование во время лабораторного практикума пакета разработки LabVIEW, однако вполне пригоден для практической апробации базовой идеи. При предлагаемом подходе в блок-диаграмме выделяется базовая Formula Node для вставки алгоритма управления, создаваемого студентом во время работы со стендом (рис. 1).

Для преобразования текста на языке Рефлекс к Си-подобному синтаксису Formula Node использовался существующий транслятор языка Рефлекс в Си, который был дополнен автоматическим преобразованием прямо задаваемых констант и констант, задаваемых через перечислитель (enum), в переменные. Также было предложено монолитное представление алгоритма, исключающее использование функций, — механизма, отсутствующего в

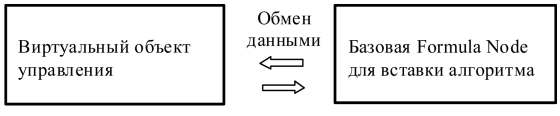


Рис. 1. Схема взаимодействия ВОУ и базовой Formula Node

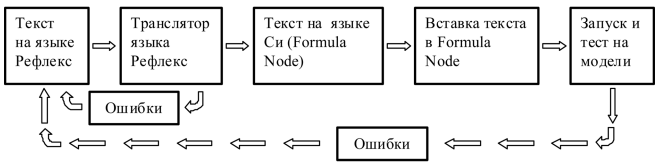


Рис. 2. Схема получения управляющего алгоритма и его отладка на модели

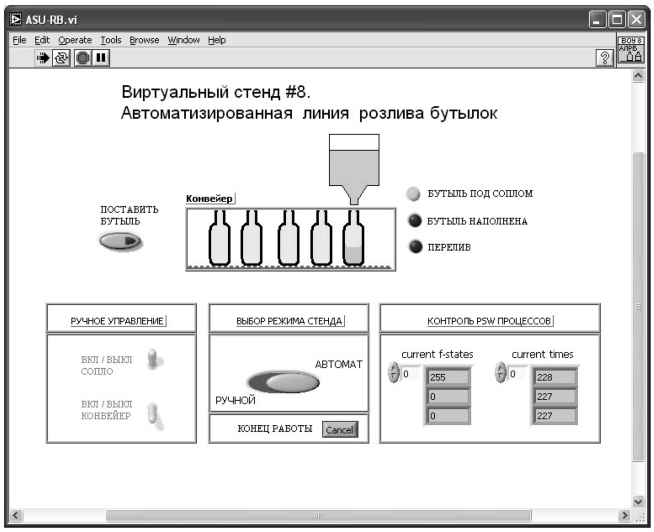


Рис. 3. Реализация виртуального стенда "Автоматизированный розлив бутылок": внешний интерфейс стенда

Formula Node. Тактирование гипераутомата (программной модели, лежащей в основе языка Рефлекс [11]) проведено штатными средствами LabVIEW: основной цикл гипераутомата "обернут" тактируемым циклом.

Подключение базовой Formula Node к проектируемому ВОУ реализовано через массивы ввода-вывода. Для автоматической выборки входных-выходных переменных из массива использован стандартный синтаксис языка Рефлекс. Внутренние переменные гипераутомата, обеспечивающие идентификацию текущих функций-состояний процессов *fcur* и текущих времен *tcur*, сохраняются в регистровых массивах, что гарантирует сохранение значений от цикла к циклу и независимость структурных связей от проектируемого алгоритма. Такой подход обеспечивает неизменность структурных связей между программными модулями для заданного виртуального стенда, поскольку позволяет полностью исключить влияние создаваемого алгоритма. Замечательно, что при предлагаемом подходе структура связей сохраняется и для различных лабораторных стендов, за исключением места присоединения модели виртуального объекта к массивам ввода-вывода, которые, очевидно, должны быть откорректированы в соответствии со сценарием задачи и моделью объекта автоматизации.

Работа студента с виртуальным стендом в рамках предлагаемой концепции происходит следующим образом (рис. 2). Сначала создается текстовое описание алгоритма на языке Рефлекс, затем запускается трансляция в Си-формат. При обнаружении ошибок во время трансляции происходит возврат к редактированию. После успешной трансляции алгоритм управления, представленный уже в Си-формате, вставляется в Formula Node. Полученная программа запускается на исполнение, и алгоритм управления тестируется на корректность путем создания различных ситуаций, предусмотренных сценарием задачи. В случае обнаружения некорректного поведения алгоритма происходит возврат к начальному этапу редактирования текста алгоритма управления.

Тестовый пример виртуального лабораторного стенда

Практическая применимость предлагаемой методики для создания виртуальных лабораторных стендов исследовалась на задаче управления автоматизированной линией розлива бутылок.

Автоматизированная линия розлива бутылок состоит из конвейера, по которому слева направо двигаются пустые бутылки, и бака с разливаемой жидкостью. Конвейер можно включать и отключать. В баке имеется сопло, которое можно открывать и закрывать. Также в системе присутствуют два дискретных датчика: для определения нахождения бутылки под соплом и контроля уровня жидкости в бутылке.

Имитационная модель линии розлива создавалась штатными средствами LabVIEW для работы с изображениями путем наложения графических примитивов конвейера и бутылки (рис. 3). Наполнение бутылки жидкостью имитировалось через постепенное закрашивание горизонтальных слоев пикселей от дна бутылки. Срабатывание датчиков наличия бутылки под соплом и уровня жидкости в бутылке определялось путем анализа цвета пикселей в за-

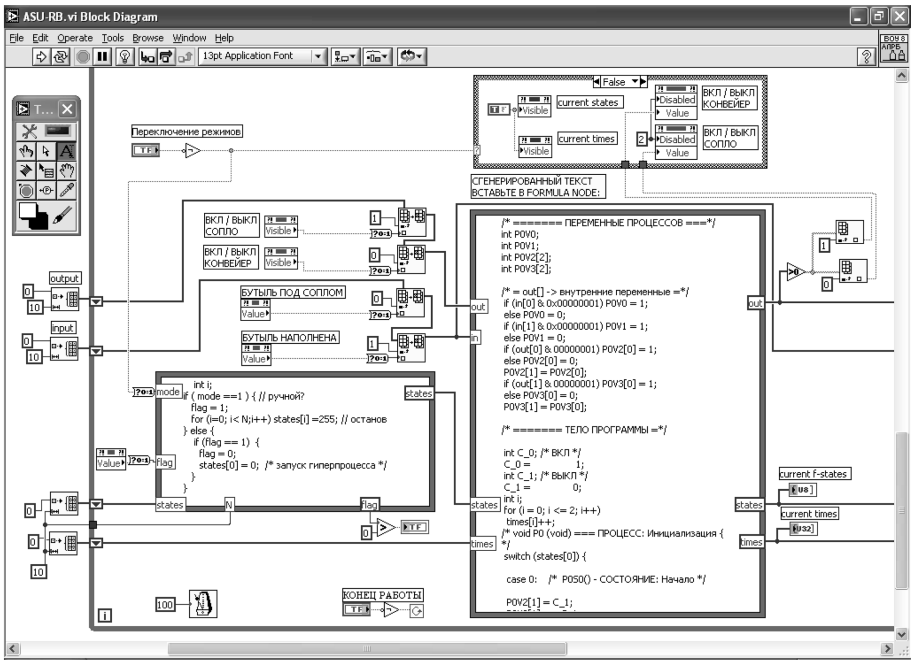


Рис. 4. Реализация виртуального стенда "Автоматизированный розлив бутылок": блок-диаграмма виртуального стенда (структурные связи Formula Node)

данных координатах сцены. В случае открытия сопла при отсутствии бутылки или при переполнении бутылки предусмотрена имитация пролива жидкости на конвейер.

Стенд предусматривает два режима управления конвейером и соплом: ручной и автоматический. В ручном режиме имеется возможность непосредственного управления линией (конвейером и соплом). В автоматическом режиме возможность ручного управления блокируется, а команды управления соплом и конвейером формируются алгоритмом, созданным студентом по описанной выше схеме. В обоих режимах имеется возможность установки бутылки на конвейер. В специальном окне отображаются числовые идентификаторы текущих функций-состояний процессов.

Фрагмент блок-диаграммы, показывающий связи базовой Formula Node для встраивания алгоритма, приведен на рис. 4.

Заключение

В работе предложена методика создания виртуальных лабораторных стендов по изучению основ программирования робототехнических комплексов. Методика основана на использовании пакета LabVIEW и языка описания управляющих алгоритмов Рефлекс.

Использование виртуальных лабораторных стендов позволяет студенту сконцентрироваться исключительно на вопросе создания алгоритма управления и упрощает изучение основ программирования ИУС. Разработанная технология интеграции гиперавтоматных блоков в среду LabVIEW может быть также использована при создании моделей объекта автоматизации для отладки реальных алгоритмов управления, а равно и для функционального расширения среды LabVIEW управляющими алгоритмами, заданными на языке Рефлекс.

Список литературы

1. **Зимин А. М., Букетин Б. В., Почуев А. П., Шумов А. В., Щепетинщиков О. А.** Учебная Интернет-лаборатория "Испытания материалов" // Информационные технологии. 2006. № 10. С. 58—65.

2. **Суворов М. В.** Автоматизированная обучающая система диспетчеров промышленной электрической сети // Промышленные АСУ и контроллеры. 2003. № 7. С. 44—45.

3. **Ершова О. В., Чистякова Т. Б.** Структура тренажерно-обучающего комплекса для процесса производства желтого фосфора // Промышленные АСУ и контроллеры. 2003. № 7. С. 46—49.

4. **Гершберг А. Ф., Подъяпольский С. В., Соркин Л. Р.** Компьютерный тренажер для обучения операторов установки каталитического риформинга в ООО "ПО "Киришинефтеоргсинтез" // Промышленные АСУ и контроллеры. 2003. № 7. С. 52—53.

5. **Ясиновский С. И.** SDBUILDER: интеллектуальная гибридная система имитационного моделирования и управления сложными дискретными системами // Автоматизация в промышленности. 2006. № 7. С. 36—42.

6. **Десятков А. Д., Сирота А. А.** Имитационное моделирование систем с адаптивной структурой на основе технологий автоматизированного создания моделей в среде MatLab + Simulink + StateFlow Учебная Интернет-лаборатория "Испытания материалов" // Информационные технологии. 2008. № 3. С. 59—66.

7. **Бутырин П. А., Васьковская Т. А., Каратаева В. В., Материкин С. В.** Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7. М.: ДМК Пресс, 2005. 264 с.

8. **Труды** VI Международной научно-практической конференции "Образовательные, научные и инженерные приложения в среде LabVIEW и технологии National Instruments", 23—24 ноября 2007. М.: Российский ун-т дружбы народов, 2007. [http://digital.ni.com/worldwide/russia.nsf/web/all/46A43B32E2F8DD50C32573E400278DD7].

9. **Вавилов К. В., Шалыто А. А.** LabVIEW и SWITCH-технология // Промышленные АСУ и контроллеры. 2006. № 6. С. 43—45.

10. **Зюбин В. Е.** "Си с процессами": язык программирования логических контроллеров // Мехатроника. 2006. № 12. С. 31—35.

11. **Зюбин В. Е.** Программирование информационно-управляющих систем на основе конечных автоматов: Учеб.-метод. пос. Новосибирск: Новосиб. гос. ун-т, 2006. 96 с. [http://reflex-language.narod.ru/index.html].

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

УДК 004.416.6

С. К. Каргапольцев, д-р техн. наук, проф., проректор по научной работе,
Иркутский государственный университет путей сообщения,
Н. В. Лашук, начальник Центра информационных технологий,
Забайкальский институт железнодорожного транспорта, e-mail: vc@zab.megalink.ru

Система поддержки принятия решений для обеспечения автоматизации управления вузом

Рассматривается применение системы поддержки принятия решений в автоматизированной системе управления вузом, что позволяет организовать корректное функционирование и взаимодействие всех подсистем автоматизированной системы управления вузом.

Ключевые слова: *система поддержки принятия решений, автоматизированная система управления вузом, хранилище данных, оперативная аналитическая обработка данных, витрина данных.*

Обоснование применения поддержки принятия решений

Одним из важнейших факторов, влияющих на обеспечение образовательной и финансово-хозяйственной деятельности вуза, является развитие единого информационного пространства на основе современных информационных технологий. Таким образом, проведение комплекса исследовательских и аналитических работ по определению структуры и методов учета первичной информации, а также анализ данных и формирование экономических показателей деятельности вуза являются актуальной задачей.

Современный уровень информационных технологий позволяет вести речь об организации автоматизированной системы управления вузом как системы поддержки принятия решений. Для решения задач автоматизации при проектировании и реализации АСУ ВУЗ нами предложено использование системы поддержки принятия решений на основе трехуровневого хранилища данных.

Система поддержки принятия решений на основе трехуровневого хранилища данных

Система поддержки принятия решений (СППР) — диалоговая автоматизированная информационная система, использующая правила принятия решений и соответствующие модели баз данных, а также интерактивный компьютерный процесс моделирования, поддерживающий принятие самостоятельных и неструктурированных решений отдельными лицами, принимающими решения [1].

Основываясь на общих принципах создания СППР, можно предложить обобщенную схему реализации АСУ ВУЗ с применением СППР (рис. 1).

Схема состоит из двух информационных уровней: подразделения и администрации. Первый информационный уровень — СППР — относится к оперативной системе, которая предназначена для оперативного реагирования на текущую ситуацию. Системы данного типа имеют название "Информационные Системы Руководства" — ИСР (*Exetcutive Information System*). К ним относятся системы, связанные с оперативным руководством организацией, системы организации документооборота (*Management Information Systems*). Наполнение баз данных на этом уровне проводится отделами вуза (данные, поступающие из приемной комиссии, факультетов, кафедр, бухгалтерии, отдела кадров и др.).

Технологию работы СППР первого информационного уровня можно рассмотреть на примере промежуточной аттестации студентов. Промежуточные данные по выполнению учебного плана студентами, посещаемость, участие в студенческой жизни института сохраняются в локальных базах данных, агрегируются и предоставляются как в печатных, так и в электронных формах лицам, принимающим решения, в данном случае — деканам факультетов. Деканы на основании полученных данных формируют приказы и распоряжения в электронном виде, которые регулируют работу факультетов. Аналогично происходит наполнение данных в других отделах и подразделениях, которые являются источником информации для обработки и агрегирования на втором уровне СППР.

Реализацию первого информационного уровня предлагается выполнить в трехуровневой архитектуре "клиент—сервер" с сервером приложений и сервером баз данных. Первый архитектурный уровень — уровень интерфейса пользователя — отвечает за представление информации пользователю и вызовы функций. Второй архитектурный уровень — уровень логики предметной области, реализующий основную функциональность приложений. Данный уровень выполняет функции сервера для интерфейса пользователя и клиента для уровня ресурсов. Третий архитектурный уровень — уровень ресурсов, на нем функционирует SQL-сервер (в нашем случае MySQL). Взаимодействие между первым и вторым архитектурными уровнями организовано с помощью технологии CGI (*Common Gateway Interface*), между вторым и третьим — с помощью языка SQL (рис. 2).

Второй информационный уровень относится к стратегической СППР (*Decision Support Systems*) и основан на анализе большого количества информации различного характера. Стратегические СППР предполагают глубокую проработку данных, которые специально преобра-

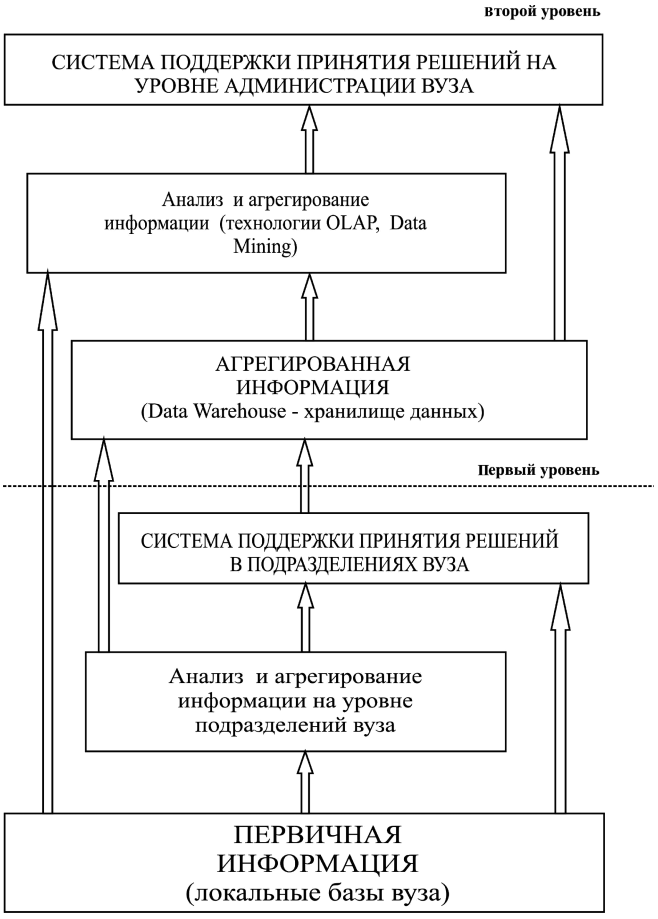


Рис. 1. Обобщенная схема системно-инструментальной поддержки автоматизации вуза

Рис. 2. Трехуровневая архитектура "клиент-сервер"

зованы таким образом, чтобы их было удобно использовать в ходе принятия решений. В АСУ ВУЗ это анализ ретроспективной информации по динамике набора студентов, выпуску по различным специальностям, финансовым затратам по различным статьям, оценка качества преподавания и другая информация, на основании которой возможна корректировка планов набора, финансового распределения средств, учебной нагрузки, хозяйственной деятельности, формирование перспективных планов развития вуза. Второй информационный уровень предполагает использование таких концепций, как хранилища данных (*Data Warehouse*), оперативная аналитическая обработка данных (*On-Line Analytical Processing, OLAP*), витрина данных — специализированное хранилище данных, ориентированное на одно из подразделений вуза (*Data Mart*).

Технологию работы СППР второго информационного уровня можно рассмотреть на примере принятия годового финансового плана. На основании ретроспективных данных, полученных из хранилища, проводится экспертная оценка сформированного плана на текущий год, после чего принимается решение (администрацией вуза) об утверждении плана или о дальнейшей доработке. Подобные экспертные оценки могут также проводиться и для планов приема, и для планов выпуска студентов, и для выявления проблемных дисциплин, для которых может потребоваться изменение учебного плана, очередности преподавания такой дисциплины.

Для реализации первого и второго информационных уровней предлагается использовать объектно-ориентированный язык программирования PERL, поддерживающий мощное средство обработки текста — регулярные выражения (*Regular Expressions*). Регулярные выражения совместно с возможностями языка SQL выполняют различные сложные выборки, фильтрацию больших объемов данных и поиск информации из хранилищ данных. Хранилище данных основано на СУБД MySQL с помощью организации структуры взаимосвязанных таблиц. Клиентская часть использует стандартные web-обозреватели (*Internet Explorer, Opera, Mozilla*). Каждое структурное подразделение имеет свой web-интерфейс, позволяющий получать сведения, на основании которых в соответствии с архитектурой АСУ принимаются решения, относящиеся как к оперативному, так и к стратегическому руководству.

Хранилище данных объединяет информацию, имеющуюся в информационных системах вуза, и подготавливает ее для анализа. Можно выделить основные проблемы деятельности вуза, анализ данных для которых будет наиболее актуален: обработка данных большого объема, разветвленная структура подразделений, требования руководителей высшего звена для более точного анализа работы вуза. Также для принятия решений необходимы отчеты, составляемые на основе различных данных из информационных систем, и быстрый доступ к информации, необходимой для анализа структурных подразделений. В отличие от оперативных систем, хранилище данных содержит информацию за весь требуемый временной интервал (вплоть до нескольких десятилетий) в едином информационном пространстве, что делает такие хранилища идеальной основой для выявления временных зависимостей и других важных аналитических показателей [3]. Информация в хранилище данных организуется в форме многомерного куба, данные предварительно агрегированы на всех соответствующих уровнях, для того чтобы обеспечить максимально быстрый доступ к ним.

Преимущества АСУ ВУЗ на основе СППР

Таким образом, АСУ ВУЗ, созданная на основе СППР с архитектурой трехуровневого хранилища данных, имеет ряд преимуществ перед ранее созданными автоматизированными системами:

- возможность выявления зависимостей и других важных аналитических показателей;
- построение отчетов на основе достоверных данных;
- своевременное обнаружение ошибок в информации, поступающей в хранилище данных;
- сравнение и интегрирование данных, вводимых и накапливаемых в различных оперативных системах;
- прогнозирование и оперативное принятие решений.

Преимущества автоматизированной системы управления вузом с применением СППР позволяют более взвешенно подходить к решению учебных, финансовых, хозяйственных вопросов с меньшей вероятностью на ошибку, оперативно реагировать на создавшуюся ситуацию, использовать минимальное время на аналитику существующих проблем, прогнозировать нежелательные ситуации. Данные преимущества также позволяют сократить вспомогательный персонал в аппарате управления за счет автоматизации сбора, обработки и дальнейшего использования информации.

Применение СППР на основе трехуровневого хранилища данных, организованного на программном обеспечении (ПО) с открытым кодом, позволило обеспечить более быстрое внедрение и корректное функционирование автоматизированной системы управления вузом.

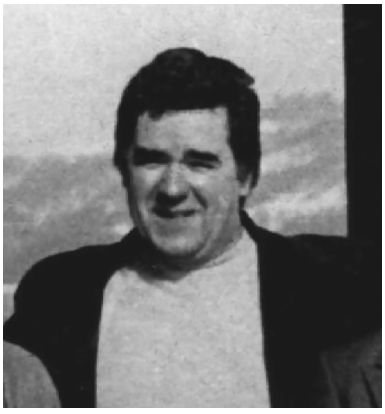
Список литературы

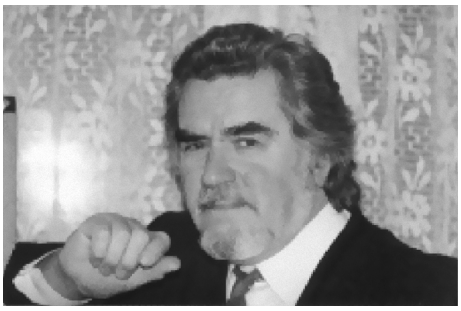
1. **Годин В. В., Корнеев И. К.** Управление информационными ресурсами: 17-модульная программа для менеджеров "Управление развитием организации". Модуль 17. М.: ИНФРА-М, 1999. 432 с.

2. **Интеграция** информационных технологий в системных исследованиях энергетики / Л. В. Массель, Е. А. Болдырев, А. Ю. Горнов и др. Под ред. Н. И. Воропая. Новосибирск: Наука, 2003. 320 с.

3. **Лисянский К.** Архитектурные решения и моделирование данных для хранилищ и витрин данных. http://www.eduh-mao.ru/.

Памяти Леонарда Андреевича Растригина (к 80-летию со дня рождения)





23 июля 2009 г. исполняется 80 лет со дня рождения известного крупного русского советского ученого и педагога в области кибернетики и информатики, доктора технических наук, профессора Леонарда Андреевича Растригина (23.07.1929—30.01.1998).

Родился Л. А. Растригин в 1929 г. в городе Елабуге (Татарстан, Россия). Учился в Москве, окончил Московский авиационный институт (МАИ) в 1953 г. по специальности "инженер-механик". Окончил аспирантуру и в 1959 г. защитил кандидатскую диссертацию. По распределению был направлен в Институт машиноведения АН СССР младшим научным сотрудником, где проработал с 1959 до 1962 г. В 1962 г. был приглашен в Ригу для организации научной лаборатории "Адаптивные системы" в создаваемом Институте электроники и вычислительной техники Академии наук Латвийской ССР, которую и возглавил. В 1966 г. в этом же институте он защитил докторскую диссертацию на тему "Статистическая оптимизация". С 1982 г. Леонард Андреевич продолжает научную и педагогическую деятельность в Рижском политехническом институте (ныне — Рижский технический университет) в должности профессора кафедры вычислительной техники.

Область научных интересов профессора Л. А. Растригина удивительно масштабна. Это и экстремальное управление, и методы адаптации сложных систем, и методы оптимизации сложных объектов; модели и методы распознавания образов, проблемы принятия решений, компьютерные методы обучения иностранным языкам с позиций управления, исследование процессов познания. Большое внимание Л. А. Растригин уделял развитию бионических подходов в технической кибернетике: бионических алгоритмов случайного поиска, в том числе эволюционных. За много лет до "западного бума" им были активно поддержаны научные исследования по эволюционным и генетическим алгоритмам.

Л. А. Растригин — общепризнанный в СССР, России и за рубежом "отец" школы случайного поиска. Пятьдесят лет назад — 17 августа 1959 г. — им были сделаны заявки на способ и устройство автоматического управления, базирующегося на процедуре случайного поиска (авторские свидетельства СССР № 129701 и № 129702). Однако автор рассматривал случайный поиск не только как технический алгоритм, но и как инструмент представления и познания природных закономерностей. Он отмечал, что "механизмы случайного поиска, по-видимому, свойственны природе нашего мира на всех уровнях его проявления и организации. И, во всяком случае, могут служить удобной и конструктивной моделью этих процессов".

Л. А. Растригин очень многое сделал для развития созданного им научного направления "Случайный поиск в задачах оптимизации". Он организовал и в течение многих лет возглавлял подкомиссию по случайному поиску комиссии "Адаптивные системы" Научного Совета по комплексной проблеме "Кибернетика" при Президиуме АН СССР. Под его патронажем возникли исследовательские коллективы, которые применяли инструментарий случайного поиска для решения актуальных задач в различных областях практики. География таких коллективов обширна — это Москва и Ленинград, Днепропетровск и Харьков, Кемерово и Красноярск, Ростов-на-Дону и Нижний Новгород, Ереван и Ташкент, Рига и Вильнюс, Йошкар-Ола и Таллин, а также многие другие города. Подкомиссией было организовано несколько десятков конференций, семинаров, рабочих встреч и школ молодых ученых в разных городах СССР, изданы десятки сборников тезисов и материалов исследований в области случайного поиска. Сам Леонард Андреевич выступил в качестве организатора и научного редактора серии сборников "Проблемы случайного поиска" (Рига: Зинатне, вып. 1—10, 1970—1982 гг.). Существенную часть его жизни занимала научно-педагогическая деятельность. Под его научным руководством было подготовлено более 70 кандидатов наук, защитили докторские диссертации 10 молодых ученых.

Леонардом Андреевичем написано 20 монографий, свыше 300 статей, 11 научно-популярных книг. Такие его монографии, как "Статистические методы поиска" (М.: Наука, 1968. 375 с), "Системы экстремального управления" (М.: Наука, 1974. 630 с.), "Современные принципы управления сложными объектами" (М.: Сов. радио, 1980. 230 с), "Адаптация сложных систем. Методы и приложения" (Рига: Зинатне, 1982. 475 с.) по праву входят в золотой фонд отечественной научно-технической литературы.

Большое место в его творчестве занимала популяризация научных знаний. Его книга "Этот случайный, случайный, случайный мир" (М.: Молодая гвардия, 1969, 1974) переведена на английский, немецкий, японский, венгерский, болгарский, эстонский и литовский языки. Книги "Кибернетика как она есть" (совместно с П. С. Граве. М.: Молодая гвардия, 1975), "По воле случая" (М.: Молодая гвардия, 1986) и многие другие, несмотря на вроде бы солидный возраст, не устарели и для современного читателя.

Леонард Андреевич был эрудитом, лидером, оптимистом, жизнелюбом, открытым и дружелюбным в общении. Это был человек-магнит для коллег и творческой молодежи. Он всегда был в центре обсуждения наиболее острых научных проблем.

Имя Ученого и Педагога Леонарда Андреевича Растригина помнят и чтят не только его друзья, коллеги, ученики и ученики его учеников, но многие ученые и специалисты как в нашей стране, так и за рубежом.

Букатова И. Л., Гринченко С. Н., Нечаев В. В.

CONTENTS

The problem of Web-services technology application for interactions support of remote private networks joined by global network is considered. An approach to solving this problem is proposed. The approach is based on establishing a protected tunnel through a global network using a special network service, which supplies facilities for server-to-server interaction, server-to-server data routing, unauthorized access protection and combining of data transfer with data processing.

Keywords: distributed systems, Internet-technologies, Web-services.

The systematization of the scientometric indicators applicable for the evaluation of scientific activity of the scientists, research institutes, areas of scientific activity and for calculations of the scientific printed products rating are carred out. The detailed characteristics of each indicator and area of its application is presented.

Keywords: scientometric indicators, valuation of scientific activity, rating to scientific printed products.

In the article from positions of biosystem analogy the methodological questions of cybernetic systems designing are considered, to which the systems of safety monitoring of information systems concern. The approach to designing the protected information systems is offered. The approach is based on analogy of architecture and mechanisms of protection of biological systems and cybernetic systems.

Keywords: biosystem analogy, information protection, intellectual systems of protection, knowledge bases, adaptive qualifiers.

A known key agreement protocol with an implicit authentication of users is analysed in the paper. It is proved that this protocol assumed earlier safe is not secure. Recommendations on improving and refining of the protocol are given.

Keywords: key agreement protocols, authentication, security analysis.

The problem of analytical support of differential diagnostic of a pathology of cervical area of a column (CAC) is esteemed in the article. The technology of detection of a pathology type and the way of forming of a structural pathology of CAC are submitted. Algorithms of informational and program support on the basis of the adapted discriminant analysis (DA) are designed and tested.

Disability of DA as a multivariate method of data analysis in problems of support of diagnostic of an orthopedic pathology without its preliminary adaptation is detected.

Keywords: the discriminant analysis, diagnostics, information-program support, a set of solving rules, resolution of conflicts, algorithm of differential diagnostics, structural infringements of cervical department of a backbone, upper cervical department of a backbone.

The possibilities of matrix system MATLAB R2008a in bioinformatics are considered. On the pattern of the analysis of gene sequences of bird flu virus the work of the advanced package Bioinformatics Toolbox with the world Internet resources in genetics is described. Besides the means of this package for the working up of microarrays and mass spectrometry data are considered. The conclusion about the possibility of using the means of bioinformatics in solving different scientific and general educated tasks is made.

Keywords: bioinformatics, sequence analysis, microarray analysis, mass spectrometry data, clustering, phylogenctic trees.

Analysis of mass testing results has revealed that the use of traditional fit statistics has proved inefficient to detect false testing results. The article highlights the findings of a research which contained a set of calculating experiments based on the Rasch model. The research has allowed to reveal the causes of the problem as well as work out a scheme of processing information to boost the efficiency of detecting false testing results.

Keywords: testing, detecting disturbances, fit statistic, normalized deviation, critical value, Rasch model, calculating experiment.

The paper presents a method of virtual educational boards creating. The educational boards are intended for practical training on applied programming of control algorithms. The education boards assume programming in the Reflex language (also known as "C with processes"). Controlled objects and their animation are simulated with the Lab-VIEW environment. The control algorithms written in the Reflex language are translated for inserting in the LabVIEW workbench via the Formula Node mechanism. The approach improves the way students learn. Rather than focusing on sometimes-tedious work with physical model of a controlled object, educators and students can focus on results and concepts of control algorithms programming.
Keywords: teaching of programming, computer simulation, control algorithms, the Reflex language.

In Kargapoltsev S. K., Lashuk N. V.'s article the application of decision support systems (DSS) realised on the software with an open code in the high school automated managerial system is considered. This direction in application DSS allows to organize correct functioning and interaction of all subsystems of the high school automated managerial system.
Keywords: decision support systems, the high school automated managerial system, data warehouse, on-line analytical processing, data mart.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**
E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *З. В. Наумова*
Сдано в набор 01.04.2009. Подписано в печать 19.05.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,78. Уч.-изд. л. 12,40. Заказ 452. Цена договорная.
Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.
Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15