

Министерство образования Республики Беларусь
Брестский политехнический институт

**«СОВРЕМЕННЫЕ ПРОБЛЕМЫ
МАТЕМАТИКИ И
ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»**

**Труды региональной конференции молодых ученых и студентов
16-18 ноября 1999 г.**

Брест 2000

ББК 32.973
УДК 681.3

Региональная конференция молодых ученых и студентов «Современные проблемы математики и вычислительной техники»/Материалы конференции.-Брест:БПИ,1999, 127 стр.

Редакционная коллегия:

1. В.Г. Федоров БПИ, Брест, ректор института, председатель
2. В.В. Бреславский БГУИР, Минск, д.т.н.
3. Р.Х. Садыхов БГУИР, Минск, д.т.н.
4. В.Н. Ярмолик БГУИР, Минск, д.т.н., сопредседатель
5. В.А. Головкич БПИ, Брест, к.т.н.
6. Н.В. Кудинов БПИ, Брест, к.т.н.
7. В.И. Хвещук БПИ, Брест, к.т.н.
8. Н.В. Лазакевич БГУ, Минск, д.ф.-м.н., сопредседатель
9. В.С. Рубанов БПИ, Брест, к.ф.-м.н.
10. А.И. Тузик БПИ, Брест, к.ф.-м.н.
11. Л.П. Махнист БПИ, Брест, к.т.н.
12. В.М. Ракецей БПИ, Брест, к.ф.-м.н.

Рецензент: А.А. Прихожий, зав. кафедрой «Программное обеспечение вычислительной техники и автоматизированных систем», БГПА.

ISBN 985-6584-08-6

ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ

Бахтизин В.В.; Перегуд С.В.

Белорусский государственный университет информатики и радиоэлектроники
220027, г. Минск, ул. П. Бровки, 6

Аннотация: Предлагается технология интеграции информационных ресурсов. Технология основывается на разработанных методах и программных средствах интеграции информационных ресурсов и классах внутреннего интерфейса Delphi.

Ключевые слова: Информация, интеграция ресурсов, информационный ресурс, информационный объект.

В настоящее время большое внимание уделяется информации, особенно ее представлению. В этом помогают информационные системы. Представление информации сильно зависит от ее вида (текст, графика или анимация). Наибольший эффект от представления информации достигается при использовании текстовых, графических, анимационных и других информационных ресурсов (ИР) путем интеграции их в единое целое.

Ознакомившись с рынком мультимедийных приложений можно сделать вывод, что существует большое количество самых разнообразных программ для работы с графикой, анимацией, звуком и другими ИР. Однако сейчас практически отсутствуют инструментарии, обладающие одновременно простотой и универсальностью для создания и интеграции ИР.

Поэтому перспективной является разработка программных продуктов, позволяющих:

- упростить, унифицировать и ускорить процесс интеграции ИР;
- расширить диапазон событий для манипулирования ИР путем создания новых событий;
- использовать возможность описания ресурсов с помощью языков высокого и низкого уровня, что значительно уменьшило бы размер ИР и увеличило скорость его отображения.

В основу предложенной технологии положены методы и программные средства системной интеграции ИР. Методы и программные средства базируются на иерархической объектной модели ИР и событийном управлении про-

цессами создания, интеграции и просмотра сценария ИР. Методы и программные средства используют внутренний интерфейс Delphi.

Интеграция информационных ресурсов подразумевает создание сценария, т.е. алгоритма, на основании которого происходит манипулирование ИР. Для работы с ИР необходимо ввести новое понятие – информационный объект (ИО). Это ресурс, который обладает набором методов и свойств необходимых для управления им во время разработки и выполнения сценария. Методы и свойства ИО делятся на методы и свойства времени разработки и времени выполнения. Методы и свойства времени разработки необходимы для отображения и манипулирования объектом при разработке сценария. Методы и свойства времени выполнения управляют ИР при запуске сценария.

Базовыми элементами, на которых основана предлагаемая технология, являются:

- иерархия классов ИР (рис.1.);
- эксперты – специальные классы, с помощью которых осуществляется взаимодействие между пользователем и средой Delphi (рис.2.).

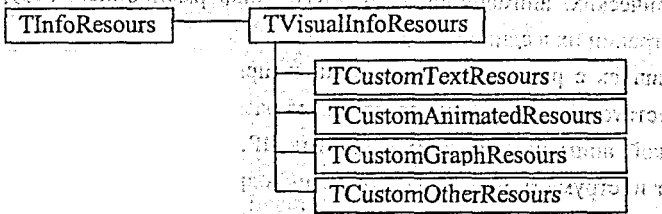


Рис.1. Иерархия классов ИР

Иерархия представленная на рис.1. необходима для описания стандартных ИР и создания новых ИР.

Класс TInfoResours содержит абстрактные методы для отрисовки и манипулирования любым ИО без привязки его к среде разработки. Класс TVisualInfoResours содержит методы и свойства для визуальной работы с ИО во время разработки сценария. Классы TCustomTextResours, TCustomGraphResours, TCustomAnimateResours инкапсулируют методы и свойства для работы с конкретным ИР, таким как текст, графика, анимация. TCustomOtherResours используется для описания нового ИР, отличного от стандартных.

Для работы с иерархией классов используются эксперты, структура которых приведена на рисунке 2. Они служат для организации удобного и простого интерфейса с пользователем.

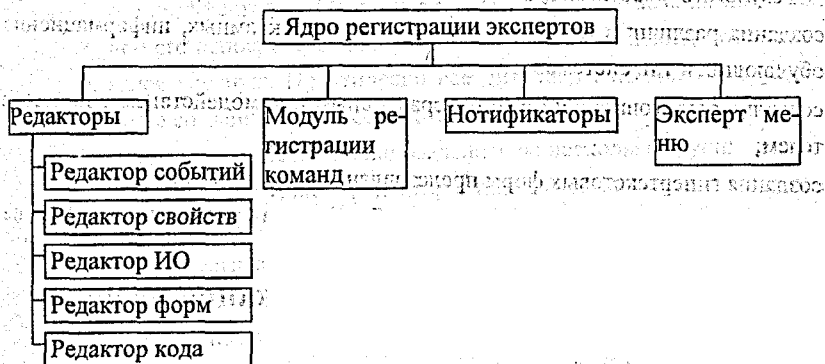


Рис.2. Структура экспертов

Ядро регистрации экспертов предназначено для установки экспертов в IDE Delphi и осуществления взаимодействия между зарегистрированными экспертами.

Редактор кода используется для отображения и редактирования исходного текста сценария информационных ресурсов. Редактор форм – для манипулирования информационными объектами во время проектирования сценария. Редакторы событий, свойств и ИО используются для удобного и быстрого доступа к свойствам и событиям выбранного ИО.

Для создания новой команды используется модуль регистрации команд, используемых для управления ИР во время выполнения сценария. Под командой понимается действие, с помощью которого можно манипулировать ИО, например, переместить ИР.

Для передачи системе информации о процессах происходящих в IDE предназначены специальные эксперты – нотификаторы.

Эксперт меню предоставляет стандартный и удобный интерфейс для взаимодействия с пользователем.

Достоинства предлагаемой технологии:

- открытая и гибкая объектная модель классов;
- возможность создания новых событий для управления ИО во время выполнения;
- интерактивное управление ресурсами во время выполнения сценария;

- использование языков высокого и низкого уровня для описания ИО;
- использование возможностей OpenGL для описания и отображения ИО.

Результаты работы могут быть эффективно использованы с целью:

- создания различных ИР в демонстрационных, рекламных, информационно-обучающих и т.п. системах;
- создания обучающих систем с интерактивным взаимодействием с пользователем;
- создания гипертекстовых форм представления информации;
- создания автоматизированных курсов обучения для различных предметных областей.

ДИЗАЙНЕРЫ SQL ЗАПРОСОВ. КЛАССИФИКАЦИЯ. ПРИНЦИП ФУНКЦИОНИРОВАНИЯ.

Соколов С.А.

Белорусский государственный университет информатики и радиоэлектроники
г.Минск, ул.Платонова 39, к.807

Аннотация: В данной статье рассмотрены средства автоматизации построения SQL запросов к базам данных. Предлагается классификация дизайнеров SQL запросов. Приведены примеры таких дизайнеров.

Ключевые слова: база данных, запрос, язык SQL, визард, дизайнер.

Язык SQL (Structured Query Language) - язык структурированных запросов, используется при создании запросов к базам данных (БД). Этот язык выступает в качестве стандарта для команд доступа и управления данными. Все современные БД (Oracle, Sybase, Informix, SQL Server, Access и др.) реализуют поддержку языка SQL. На основе языка SQL строятся SQL запросы к БД.

Запрос – это логически законченная команда по управлению данными. Запросы могут выполнять следующие операции: создавать таблицы, определять уникальные по значению поля, образовывать связи между таблицами, добавлять, изменять и удалять записи (строки, картежи) в таблицах. По мере роста сложности операции над данными растет объем SQL запроса. Размер большого запроса может достигать страницы текста, состоящей из команд языка SQL. Поэтому для автоматизации создания SQL запросов применяют визарды.

Визарды – это средства автоматизации выполнения какой-либо операции. Визарды чаще именуются дизайнерами (от англ. design – строить). Дизайнерами

SQL запросов позволяют автоматизировать процесс создания запроса на языке SQL.

Безусловно, создавать запросы вручную можно в обычном редакторе, но это длительно и неэффективно. Отладка запроса может занимать намного больше времени, чем его непосредственное создание. Никто не оценит, что сложный запрос создавался вручную. Практически все дизайнеры позволяют осуществлять тестирование созданных запросов – отладку.

Формально все дизайнеры можно разделить на несколько групп:

1. Дизайнеры текстового типа – редакторы позволяющие создавать текст запроса в обычном окне редактирования (консоли). К дизайнерам такие редакторы можно отнести лишь основываясь на значении англоязычного слова design (designer), что означает строить (средство построения). Строить запросы можно и с помощью консоли – самого древнего компьютерного способа.

Отладки запроса как таковой не существует. Запуск запроса на исполнение осуществляется при выборе соответствующего пункта меню или специальной кнопкой в панели инструментов. Если в тексте запроса присутствует синтаксическая ошибка, то при запуске запроса на исполнение будет выдаваться соответствующее сообщение.

Достоинства – простота оболочки дизайнера. Недостатки - сложность поиска логических ошибок. Ошибок более сложных, чем несовпадение типов или нарушение целостности. Обычно применение данного типа дизайнеров аргументируют следующим образом: разрабатывая запросы вручную, можно добиться их максимальной эффективности. Пример таких редакторов вы можете встретить в оболочке любой современной СУБД.

2. Визарды – это класс программ ориентированных на работу с базами данных. Эти программы обычно осуществляют генерацию и исполнение SQL запросов направленных на просмотр, добавление, удаление, изменение данных в базе. Визарды в свою очередь делятся на 2 группы: а) специализированные б) универсальные.

Специализированные визарды ориентируются исключительно только на определенную структуру БД и позволяют обычно осуществлять операции над данными по заранее определенному способу. Обычно в таких визардах "защиты" фиксированные запросы, результат действия которых изменяется

значениями подставляемых параметров в запрос. По своей сути клиентская

программа любой БД – это сложный специализированный визард.

Универсальные визарды предназначены для более широкой гаммы выполняемых функций и не привязываются к структуре конкретной БД. Такие дизайнеры обычно считают структуру базы (таблицы, поля, связи) и предлагают создавать запрос на основании каких-либо форм-шаблонов. Обычно все универсальные дизайнеры строятся на основании одного из следующих типовых решений:

а) Применение многосвязных списков – формы, содержащей несколько определяющих списков для конструируемого запроса: список существующих таблиц с доступом к полям таблиц (возможен еще один список полей текущей таблицы), список полей, образующих связи между таблицами, список или таблицу условий отбора. В ряде решений два последних списка объединяются в одну таблицу.

б) Использование MDI-формы – формы, содержащей следующие клиентские окна: окна таблиц (название окна – название таблицы, в окне находится список с полями), окно связей, окно с таблицей условий отбора.

в) Применение дерева таблиц с вложенными полями – дизайнер созданный с делением формы пополам на определяющую часть (ресурсы) и информационную часть (свойства). Пример интерфейса такого типа – проводник Windows.

г) Построение запроса на основе ER-диаграмм – дизайнеры профессионального типа, позволяющие вести разработку и отладку вложенных запросов с применением динамических таблиц.

В качестве примера можно привести дизайнер запросов, поставляемый совместно с редактором географических карт WinCAD фирмы Siemens.

Это самый удачный представитель универсальных дизайнеров известных мне.

3. QBE (Query by Example – запрос по примеру). Из названия понятна идея построения запроса, заложенная в QBE. На самом деле это универсальный визард. Вынести его в отдельную группу необходимо из-за его специфических свойств. Это лишь один метод, который признан как стандарт построения запросов на основе шаблонов.

Шаблон выборки содержит следующие информационные строки:

- а) поле – название поля, над которым осуществляется соответствующая операция,
- б) таблица – таблица в которой находится данное поле (в запросе могут использоваться несколько таблиц),
- в) сортировка – она может быть по возрастанию, по убыванию, либо не использоваться,
- г) вывод на экран – содержит чекбокс (если включен - поле показывается, в обратной ситуации - нет),
- д) строки условий отбора (3-7 строк) – могут содержать логические операции типа =, <, >, <=, >=, <> или вложенные запросы.

Примерами могут послужить широко известные представители MS Office – MS Query и Designer MS Access.

Литература

1. Соколов С.А. Сборник тезисов студенческой научно технической конференции. Вертикальный QBE. Мн., ротопринт. Белорусский государственный университет информатики и радиоэлектроники, 1999.
2. Бойко В.В. Организация и использование баз данных. –М., Мир, 1991.
3. Грабер М. Введение в SQL. –М., Мир, 1996.

АЛГОРИТМИЧЕСКАЯ МУЛЬТИЯЗЫЧНОСТЬ. ПРОБЛЕМЫ. ПУТИ РЕАЛИЗАЦИИ.

Соколов С.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, ул. Платонова 39, к.807

Аннотация: В данной статье рассмотрены проблемы алгоритмической мультязычности и пути реализации. Приведены основные причины тормозящие развитие мультязычности.

Ключевые слова: язык программирования, компилятор, интерпретатор, мультязычность.

Алгоритмическая мультязычность – это свойство компилятора или интерпретатора, заключающееся в поддержке процесса разработки программного обеспечения (ПО) с применением более одного языка программирования.

Причина возникновения нового интереса к проблеме алгоритмической мультязычности обусловлена накоплением большого объема профессиональных библиотек к столь не популярным сейчас COBOL, FORTRAN, ALGOL.

Существует масса других умирающих языков программирования. Часть ПО разрабатывалась для «больших» машин, которые уже исчезли из компьютерных центров.

Для Windows наиболее популярными языками программирования можно считать C/C++, PASCAL, BASIC и JAVA. Для Unix в наибольшей степени характерны C/C++, PERL, FORTRAN, JAVA. Медленно стал появляться интерес к языку COBOL.

Подобная идея по созданию алгоритмической мультязычности возникала в 60-х годах, но дальше конвертаторов типа C в PASCAL не пошла. Сказались следующие основные причины:

1. Не появилось достаточно удачного коммерческого продукта, который бы агрессивно внедрялся и рекламировался на рынке программного обеспечения так, как Visual C++ от Microsoft или Java от Sun. В качестве языкового стандарта со временем стал выступать язык C/C++. Если подробно изучать синтаксис большинства языков программирования, созданных после создания C/C++, то очень четко просматривается попытка создания очередного клона C/C++.
2. Нежелание крупных фирм терять возможность поставки пользователям нескольких продуктов вместо одного универсального. Это бы резко снизило прибыль от продаж и еще более усилило конкуренцию на рынке программного обеспечения.
3. Сложность реализации эффективного алгоритма и наличие большого количества клонов с несовместимыми языковыми расширениями. Обычно фирмы стремятся поддерживать только синтаксис самого языка. В остальном фирмы руководствуются собственными представлениями о программировании.
4. Сложность совместимости типизируемых языков программирования с не типизированными. Например, BASIC был изначально создан как не типизируемый язык (Visual Basic имеет типизируемые переменные), а PASCAL исключительно как типизируемый.
5. Появление новой модели развития коммерческих продуктов. Вместо поддержки нескольких языков программирования на одной платформе в одном средстве предлагается вариант одного языка программирования для разных платформ. Эту модель используют для продвижения на рынке языков C/C++ и JAVA.

После рассмотрения причин, попробуем сформулировать основные рекомендации к будущему универсальному языковому транслятору:

1. поддержка максимально возможного количества алгоритмических языков;
2. полная поддержка объектно-ориентированного программирования;
3. возможность наращивания типов, конвертируемых программных конструкций.

Реализация алгоритмической мультязычности фактически сводится к созданию промежуточного языка. Применение этого промежуточного кода резко упростило бы поддержку нескольких языков программирования, т.к. текст программ конвертировался в промежуточный язык или код. После конвертации имелась бы возможность обработки только этого кода. Подобный подход использовался разработчиками JAVA.

Возможны несколько основных способов поддержки мультязычности:

1. автоматическое распознавание лексем по ходу анализа кода программы
2. обработка опций языкового блока (команды – старт/финиш нового языка)
3. определение принадлежности лексемы к языку определяется расширением файла

При реализации конвертаторов необходимо учитывать особенности конкретного языка программирования. С проблемой несовместимости правил различных языков программирования сталкиваешься при попытке совмещения разноязычных фрагментов программ. Некоторые основополагающие правила одного языка оказываются в противоречии с правилами другого.

Наиболее типичные различия языков программирования:

1. возможности совпадения названий файлов и классов в них;
2. поиск файлов библиотек;
3. распознавание больших и строчных символов;
4. реализация обработки ошибок;
5. особенности отладчика и его опции.

Сложности возникают с декларациями переменных и комментариями, с их выравниванием по отношению к коду. В ряде случаев при конвертации происходит искажение их расположения по отношению к коду.

1. Гришук И.П. Компиляторы. —М., Высшая школа, 1992.
2. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. —М., Издательство «МИР», 1975.

ОРГАНИЗАЦИЯ ГРУППОВЫХ ЗАПРОСОВ К МНОГОТАБЛИЧНЫМ ДАННЫМ

Галерштейн В.Е., Ревотюк М.П.

Белорусский государственный университет информатики и радиоэлектроники
Минск, ул. П.Бровки, 6, кафедра ИТАС

Предлагается метод оптимизации групповых запросов к многотабличным данным, обсуждаются условия его применимости и оценка выигрыша в скорости доступа по сравнению с традиционными подходами. Рассматриваются способы его реализации при разработке новых СУБД.

Ключевые слова: базы данных, методы доступа к данным, оптимизация запросов.

В реальных системах обработки информации часто возникает задача выборки данных из нескольких таблиц данных, связанных по некоторому ключу. Например, в системах семейства xBase подобная связь устанавливается оператором SET RELATION, а в SQL - системах - предикатом выражения WHERE оператора SELECT [8,10]. Организация выборки записей из группы таблиц наиболее эффективно строится, как правило, на основе индексных файлов. Каждой таблице ставится в соответствие один индексный файл.

Известно, что наиболее распространенная организация индексных файлов - деревья Байера (B+ деревья) [2,3,6]. Среднее время доступа T к искомой записи в B+ дереве [3,4]:

$$T = \log_a n \quad (1)$$

где n - число записей таблицы, a - число индексов на одной странице B+ дерева.

Если применять традиционную схему, последовательной обработки всех M связанных таблиц, то среднее время обработки $T(M)$ составит:

$$T(M) = \sum_{i=1}^M \log_a n_i \quad (2)$$

Логарифмический закон времени доступа к индексированной таблице (1) побуждает предложить следующую схему организации групповых запросов. Построим индексный файл, где выражения ключа включают ссылку на идентификатор таблицы, а также флаг уникальности ключа для данной таблицы:

$$K_c = K + \text{TableID}(i) + \text{Unicue}(i), i = 1..M \quad (3)$$

где K_c - выражение ключа групповой связи таблиц, $\text{TableID}(i)$ - идентификатор таблицы, $\text{Unicue}(i)$ - флаг уникальности ключа.

Так как количество отображаемых записей в групповом индексном файле

$$N = \sum_{i=1}^M n_i \quad (4)$$

а схема его организации - В+ дерево, то среднее время доступа:

$$T'(M) = \log_a \sum_{i=1}^M n_i \quad (5)$$

Преимущество введения группового индексного файла объясняется следующим из (2) и (5) выражением:

$$\prod n_i > \sum n_i, i = 1 \dots M \quad (6)$$

Обсудим вопрос реализации групповых запросов по предложенной схеме.

Например, в системе FOXPRO, являющейся типичным представителем xBase систем, реализация (3) может выглядеть так:

* СОЗДАНИЕ БАЗ ДАННЫХ С КЛЮЧЕМ

```
CREATE TABLE T1 (K C(10), D1 N(16,2))
```

```
CREATE TABLE T2 (K C(10), D2 C(80))
```

```
CREATE TABLE TM (K C(10), DM D(8))
```

* ИНДЕКСИРОВАНИЕ БАЗ ДАННЫХ:

```
FOR I=0 TO M
```

```
  PRIV TABLEID
```

```
  TABLEID='T'+STR(I)
```

```
  SELECT (TABLEID)
```

```
  INDEX ON K+TABLEID TAG KC ADDITIVE
```

```
END FOR
```

Операция создания и индексирования баз данных может выполняться и операторами языка SQL [9].

Индексные файлы T1.CDX ... TM.CDX далее требуется слить в единый файл, что в системе FOXPRO не предусмотрено, но подобная задача разрешима при программировании средствами более низкого уровня, причем ее вычислительная сложность не превосходит $N \log_2 N$ [1,3,6].

Выборка данных, в общем случае, требует обеспечения механизма обращения к записям отдельных таблиц с форматами различных СУБД. Эта технологическая задача в настоящее время тривиальна и может быть решена с помощью таких стандартизованных средств доступа, как ODBC, DAO, IDAPI, OLE DB [9]. Например, низкоуровневые средства работы с реляционными базами

данных, удовлетворяющие стандарту ISSAM (INFORMIX C ISSAM, Btree Filer Turbo Power, POET Object Database System) [7,10] реализуют следующую технологию работы с ключами и данными:

```
DBRes GetKey(DBFile file, KeyRec& Key); //Выборка ключа записи по маске
```

```
DBRes GetRecord(DBFile, KeyRec Key, DataRec& Rec); // Выборка по ключу
```

Используя подобные операции, не составляет труда последовательно получить все записи ключей из каждого индексного файла и переслать их в групповой индексный файл, где ключ записи определяется (3).

Очевидно, что современные СУБД не обладают средствами поддержки групповых индексных файлов. Однако при разработке новых систем обработки данных, например, в среде C++, реализация такого механизма вполне реальна.

Рассмотрим случай параллельной работы приложения с групповым индексным файлом и индексным файлом одной из исходных баз данных. Этот случай может иметь место при выполнении операций удаления, добавления или модификации данных в одной из исходных баз данных. Очевидно, что при использовании одиночного индексного файла подобные запросы будут выполняться быстрее, чем при использовании группового индексного файла. В современных СУБД решение о стратегии доступа принимает оптимизатор запросов. Для обеспечения корректности ключей в групповом индексном файле, оптимизатор запросов может породить фоновую задачу модификации группового индексного файла. Это устранил замедление выполнения запроса к одиночной базе данных.

Литература:

1. Атре Ш. Структурный подход к организации баз данных — М.: Финансы и статистика, 1983.
2. Вирт Н. Систематическое программирование — М.: Мир, 1977.
3. Вирт Н. Алгоритмы + структуры данных = программы — М.: Мир, 1985.
4. Дж. Ульман Базы данных на Паскале — М.: Машиностроение, 1990.
5. Дж. Ульман Основы систем баз данных — М.: Мир, 1980.
6. Кнут Д. Искусство программирования для ЭВМ. Т.3. — М.: Мир, 1978.
7. Smith J.M., and Smith, D.C.P. Database abstractions: Aggregation/Commun. ACM, 20, June, 1997.

7. Smith David, Post-Relational Database: Revitalizing Relational Technology for New Applications. - IDC White Paper, March 1994.
8. ANSI X3.135-1992, American National Standard for Information Systems - Database Language - SQL, November, 1992.
9. ANSI/X3/SPARC DMBS Framework. Report of the study group on data base management systems Inform. Systems, 1978.

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАБОТЫ С ИЗОБРАЖЕНИЯМИ

Молочко Д. А., Муха В. С.

Белорусский государственный университет информатики и радиоэлектроники
Минск, ул.П.Бровки,6, e-mail: mukha@gw.bsuir.unibel.by

В работе описывается разработанный авторами программный модуль для языка BORLAND PASCAL 7.0, предназначенный для работы с графическими файлами формата PCX.

Ключевые слова: анализ и обработка изображений, программные средства.

Задача математической обработки изображений является весьма актуальной; о чем свидетельствуют многочисленные публикации [1-4]. В работе описывается разработанный авторами программный модуль для работы с графическими файлами формата PCX. В сравнении с другими программами по обработке изображений данная разработка обладает следующими особенностями.

1. Предназначена для исследователей, занимающихся разработкой и исследованием *собственных* алгоритмов обработки случайных изображений и полей.
2. Рассчитана на пользователей, *не имеющих глубоких знаний в области форматов графических файлов*, перед которыми стоит задача с помощью общедоступных и простых средств (компилятор языка BORLAND PASCAL 7.0) обрабатывать вводимые изображения и сохранять полученные изображения.
3. Ориентирована на использование *многомерно-матричного подхода* [4] к обработке случайных изображений и полей, когда изображение представляется двумерной матрицей, а его ковариационная матрица – четырехмерной матрицей.

Разработаны и реализованы программно с использованием предложенного модуля новые алгоритмы зашумления изображений и пропускания изображений через систему линз. Модуль и программа могут быть предложены

пользователю, занимающемуся разработкой и исследованием алгоритмов фильтрации и фокусировки изображений и полей.

Программный модуль выполнен в виде модуля языка программирования PASCAL по аналогии с модулем CRT. Он подключается к разрабатываемой программе пользователя командой языка PASCAL `uses pscimage;`

Разработанный программный модуль предоставляет объекты для работы с памятью, с графическими файлами формата PCX и просмотра изображения.

Для работы с памятью выделены следующие объекты:

1. **Matrix2Real** – объект для работы с двумерной матрицей, содержащей вещественные числа.

Этот объект имеет следующее описание:

```
type Matrix2Real = object
  Values : ^Matrix2r; {указатель на массив указателей, которые указывают на одномерные массивы чисел типа real}
  n, m : integer; {n x m – размерность матрицы}
  procedure Create (n, m : integer); {процедура создания двумерной матрицы размерностью n x m}
  procedure Free; {процедура освобождения памяти, занимаемой двумерной матрицей}
end;
```

2. **Matrix2Byte** – объект для работы с двумерной матрицей, содержащей числа типа byte и имеющий следующее описание:

```
type Matrix2Byte = object
  Values : ^Matrix2b; {указатель на массив указателей, которые указывают на одномерные массивы чисел типа byte}
  n, m : integer; {n x m – размерность матрицы}
  procedure Create (n, m : integer); {процедура создания двумерной матрицы размерностью n x m}
  procedure Free; {процедура освобождения памяти, занимаемой двумерной матрицей}
end;
```

3. **Matrix4Real** – объект для работы с четырехмерной матрицей, содержащей вещественные числа и имеющий следующее описание:

```
type Matrix4Real = object
```



```

Values : ^Matrix4r; {указатель на массив указателей, которые указы-
вают на одномерные массивы указателей, указывающих на ... на одно-
мерные массивы чисел типа real}
n,m,k,l : integer; {n x m x k x l - размерность матрицы}
procedure Create (n, m, k, l : integer); {процедура создания четырехмер-
ной матрицы размером n x m x k x l}
procedure Free; {процедура освобождения памяти, занимаемой четы-
рёхмерной матрицей}
end;

```

Для того, чтобы использовать данные объекты, их нужно объявить в разделе переменных, например:

```

Var MyMatrix2 : Matrix2Byte;

```

```

MyMatrix4 : Matrix4Real;

```

Далее в программе их нужно проинициализировать :

```

MyMatrix2.Create (12, k);

```

```

MyMatrix4.Create (5, 9, 10, 80);

```

Теперь их можно использовать, например, занести другие значения:

```

For i:=1 to 5 do

```

```

  For j:=1 to 9 do

```

```

    For k:=1 to 10 do

```

```

      For l:=1 to 80 do

```

```

        MyMatrix4.Values^[i]^j^[k]^l := random;

```

Когда массивы не нужны, их нужно удалить из памяти следующим образом:

```

MyMatrix4.Free;

```

```

MyMatrix2.Free;

```

После удаления массивов обращаться к ним НЕЛЬЗЯ!

Для работы с графическими файлами формата PCX используется модуль TRCXImage, описание которого приводится ниже.

```

type TRCXImage = object

```

```

  Red : Matrix2Byte; {Красная составляющая.}

```

```

  Green : Matrix2Byte; {Зеленая составляющая.}

```

```

  Blue : Matrix2Byte; {Синяя составляющая.}

```

```

  Error : integer; {Ошибка.}

```

```

  NumOfColor : longint; {Кол-во цветов.}

```

maxX, maxY : integer; {Размер изображения.}

constructor Create (FileName : string); {Конструктор, считывает с диска файл с именем FileName и разбивает его на цветовые составляющие}

destructor Free; {Деструктор, освобождает память, занятую

цветовыми составляющими}

function SaveToFile (FileName : string; colors : integer) : integer; {Функция сохранения изображения в файл}

function ViewImage : integer; {Функция просмотра изображения}

function GetColor (i, j : integer) : TColor; {Функция получения цвета пиксела с координатами i и j}

procedure SetColor (i, j : integer; color : TColor); { Функция установки цвета пиксела с координатами i и j }

end;

Цвет пиксела задается следующим образом:

Tcolor = 00RRGGBB, где 00 – два нуля, R, G, B – шестнадцатиричные цифры.

Для создания экземпляра данного класса нужно сделать следующее:

```
Var MyImage : TPCXImage;
```

Далее в программе

```
MyImage.Create ("c:\pictures\mypicture.pcx");
```

```
If MyImage.Error <> NO_ERRORS then
```

```
Begin
```

```
Writeln ('ошибка!!!'); halt (-1);
```

```
End;
```

Теперь можно обрабатывать изображение.

Сначала красную составляющую :

```
MyImage.Red.Value^[i]^j := ...
```

Потом зеленую :

```
MyImage.Green.Value^[i]^j := ...
```

И наконец синюю:

```
MyImage.Blue.Value^[i]^j := ...
```

После обработки изображение можно сохранить :

MyImage.SaveToFile ("c:\pictures\new.pcx"; colors_16mil);

Если изображение не нужно, его можно удалить из памяти следующим образом

MyImage.Free;

После удаления изображения обращаться к нему НЕЛЬЗЯ!

Функции SaveToFile и ViewImage возвращают код ошибки, которая может при-

нимать одно из следующих значений:

FILE_NOT_FOUND – Файл не найден

FILE_IS_CREATE – Файл уже существует

CANT_CREATE – Не могу создать файл

IT_IS_NOT_PCX_FILE – Файл не PCX формата

NOT_ENOUGH_MEMORY – Не хватает памяти

UNKNOWN_FORMAT – Неизвестный формат файла

NO_ERRORS – Нет ошибок

FILE_WAS_NOT_LOADED – Изображение не загружено

DEVICE_ERROR – Ошибка устройства

Литература

1. Proceeding of Fifth International Conference PRIP'99 "Pattern Recognition and Information Processing". (18-20 May 1999, Minsk, Republic of Belarus), Volume 1. Minsk-Szczecin, 1999.
2. Прэтт У. Цифровая обработка изображений. Кн. 1. М.: Мир, 1982. - 312 с.
3. Прэтт У. Цифровая обработка изображений. Кн. 2. М.: Мир, 1982. - 792 с.
4. Муха В.С. Многомерно-матричная технология в теории моделирования изображений/ Материалы международной научно-технической конференции "Новые информационные технологии в науке и производстве". - Мн.: Белорусский государственный университет информатики и радиоэлектроники, 1998. - С. 199-202.

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АНАЛИЗА ПРОИЗВОДСТВЕННО-ХОЗЯЙСТВЕННОЙ ДЕЯТЕЛЬНОСТИ ПРЕД- ПРИЯТИЯ И ФИНАНСОВОГО ПЛАНИРОВАНИЯ

Валиев Д.А.

Белорусский государственный университет информатики и радиоэлектроники
Минск-27, ул. Бровки, 4

Аннотация: рассмотрен пакет прикладных программ для анализа производственно-хозяйственной деятельности предприятия и финансового планирования

Ключевые слова: анализ производственно-хозяйственной деятельности, финансовое планирование

Анализ производственно-хозяйственной деятельности предприятия помогает руководству видеть реальное положение дел в сфере производства, инвестирования и финансирования, а также позволяет оценить текущее состояние предприятия и перспективы его дальнейшего развития. Результаты анализа должны учитываться высшим руководством и руководителями финансовой службы при принятии решений в отношении максимизации прибыли предприятия. Для стабильной и долгосрочной работы предприятия помимо получения прибыли необходимо и поддержание ликвидности. Это означает платежеспособность предприятия в каждый рассматриваемый момент. Предприятие платежеспособно только в том случае, если сумма производимых в каждый момент выплат не превышает сумму приходящихся на этот момент поступлений и остатка переходящих платежных средств. Решения, связанные с достижением цели поддержания ликвидности предприятия могут приниматься с помощью методов планирования и прогнозирования будущего потока платежей. В этом случае задачей высшего руководства предприятия является прогнозирование степени достижения поставленной цели при использовании различных стратегий, т.е. оно должно предвидеть, какими будут в каждом случае баланс и другие финансовые показатели.

Финансовое планирование само по себе может быть полезно для моделирования будущих результатов деятельности, если менеджеры сделают некоторые предположения о будущем. Разработанные прогнозные финансовые документы делают их привлекательными в качестве приблизительных оценок, которые можно исправлять и уточнять, если есть дополнительная информация и ин-

туитивные предчувствия, основанные на опыте, особенно когда число рассматриваемых альтернатив ограничено.

Для эффективной организации работы по планированию прибыли и поддержанию ликвидности можно использовать общефирменную модель предприятия, которая преобразовывается в модель принятия решений в случае ввода целевой функции. С помощью общефирменной модели руководство предприятия может выработать долгосрочную стратегию развития, произвести необходимые расчеты ориентированные на получение максимальной прибыли и поддержание необходимого уровня ликвидности предприятия, выработать финансовую политику и спрогнозировать будущие результаты работы.

Управленческие решения можно принимать опираясь с одной стороны на предложенные альтернативы, которые разработаны с помощью модели предприятия и с другой, на результаты анализа производственно-хозяйственной деятельности.

Анализ производственно-хозяйственной деятельности предприятия имеет еще одну привлекательную сторону - результаты анализа легко интерпретируются и понятны высшему руководству.

При анализе производственно-хозяйственной деятельности предприятия и финансовом планировании приходится обрабатывать большие объемы информации и рассчитать большое количество показателей. Кроме этого высока сложность расчетов и существуют жесткие требования к срокам получения результата анализа. Информация должна быть оперативной, чтобы не потерять свою ценность. Поэтому целесообразно применение пакетов прикладных программ специализированного назначения, автоматизирующих основные этапы и процедуры исследования. Применение пакета прикладных программ позволяет более оперативно и точно рассчитывать показатели.

Прикладная программа для анализа производственно-хозяйственной деятельности должна быть гибкой, расширяемой, иметь мощные генератор отчетов, возможность импорта и экспорта данных, реализации собственных методик для нестандартных задач.

На сегодня среди высшего руководства и специалистов финансовых отделов предприятий самым популярным программными средствами являются табличный процессор MS Excel, который предоставляет средства для вычислений, статистической обработке данных и выводе результата в произвольном виде, в том числе и в графическом формате и система управления базами данных MS

Access. Но, при использовании только MS Excel и MS Access аналитику приходится выполнять много ручной работы по реализации выбранной для анализа методики. Поэтому для оптимальной работы аналитика требуется специализированный пакет прикладных программ.

Автором предлагается пакет прикладных программ для анализа производственно-хозяйственной деятельности предприятия, реализованный на базе MS Excel и MS Access. Программа позволяет выполнить комплексную оценку деятельности субъекта хозяйствования и выявить основные тенденции его развития. Основным источником информации для анализа производственно-хозяйственной деятельности служит бухгалтерская отчетность (баланс и отчет о прибылях и убытках). Кроме того, руководители предприятия, обладающие всей полнотой информации, могут использовать в анализе данные о кадрах, производстве, инвестициях и финансировании. После сбора всей необходимой информации аналитик с помощью программы имеет возможность провести анализ изменения статей и структуры баланса, показателей финансового состояния, прибыльности, оборачиваемости и рентабельности.

В программе реализованы основные методы проведения финансового анализа - горизонтальный анализ, при котором показатели сравниваются с аналогичными за другие периоды и вертикальный анализ, при котором исследуется структура показателей.

Открытость и адаптивность прикладной программы предоставляет пользователю возможность самостоятельно вносить изменения в программу, учитывая свои требования или новую методику анализа.

Литература:

1. Ковалев В.В. Финансовый анализ: Управление капиталом. Выбор инвестиций. Анализ отчетности. - 2-е изд. - М.: Финансы и статистика, 1998. - 512 с.
2. Шеремет А.Д., Сайфулин Р.С. Методика финансового анализа. - М.: ИНФРА-М, 1996. - 176 с.
3. Хил Лафуенте А.М. Финансовый анализ в условиях неопределенности: Пер. с исп. Под редакцией Е.И. Велеско, В.В. Краснопрошина, Н.А. Лепешинского. - Мн.: Тэхналогія, 1998. - 150 с.
4. Хелферт Э. Техника финансового анализа/ Пер. с англ. под ред. Л.П. Белых. - М.: Аудит, ЮНИТИ, 1996. - 663 с.
5. Савицкая Г.В. Анализ хозяйственной деятельности предприятия: 2-е изд., перераб. и доп. - Мн.: ИП "Экоперспектива", 1997. - 498 с.

ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ БИЗНЕС-ПЛАНИРОВАНИЯ

Валнев М.А., Железко Б.А.

Белорусский государственный экономический университет

Минск-70, пр. Партизанский 26

Аннотация: рассмотрена система поддержки принятия решений для бизнес-планирования. Описана ее структура и аналитическое наполнение с использованием математического аппарата.

Ключевые слова: бизнес-планирование, система поддержки принятия решений.

Большинство решений мы принимаем не задумываясь, так как существует автоматизм поведения, выработанный многолетней практикой. Но есть проблемы выбора, решая которые, человек испытывает мучительные раздумья, как правило, эти проблемы имеют исключительный не повторяющийся характер и связаны с рассмотрением целого ряда альтернатив. В таких проблемах новым является либо объект выбора, либо обстановка, в которой совершается выбор. С подобным вопросом выбора сталкивается инвестор в процессе управления инвестиционным процессом, где на рассмотрение предлагается большое количество вариантов решения задачи. И от результата выбора зачастую зависит получат ли предприятие ожидаемый эффект от осуществляемых им действий.

Для решения задач выбора руководителям высшего и среднего управленческого звена следует использовать систему поддержки принятия решений (СППР). СППР предназначена для обработки данных и реализации моделей, помогающие решать отдельные, в основном слабоструктурированные задачи (например, решение о займе или инвестициях, составление прогнозов и т.п.). Благодаря модельному принципу реализации система способна предоставить аналитику, в процессе бизнес-планирования, возможность проводить анализ в условиях приближенных к реальности. Приближение достигается за счет учета постоянной изменчивости окружающей среды. Это может быть изменение цен на рынке из-за воздействия конкуренции, изменение издержек на продукцию под влиянием внешней среды, изменение процентных ставок на инвестиционные ресурсы и др. Система дает, в процессе анализа, возможность комбинировать (изменять с учетом изменения окружающей среды) входные данные, что способствует проведению более углубленного анализа.

Это позволяет аналитику принимать решение на основе рассмотрения всех возможных альтернатив реализации инвестиционного процесса.

СППР призвана обеспечить руководителей информацией, необходимой для принятия индивидуальных или групповых решений тактического плана. Обычно такие решения имеют важное значение на определенном временном интервале (декада, месяц, квартал). Потребность в автоматизации данного процесса, как свидетельствует практика многих стран в будущем будет возрастать. Темпы внедрения подобных технологий в Беларуси сдерживаются значительным дефицитом финансовых средств и отсутствием достаточного опыта использования информационных технологий в бизнесе.

Автоматизация обработки исходных данных реализуемая в СППР позволяет качественно поднять уровень принимаемых решений. Суть автоматизации заключается в известном высказывании "время деньги". Благодаря автоматизации аналитик способен обработать огромный поток информации, которая характеризует как внешнюю, так и внутреннюю среду. Снижая при этом затраты на ее обработку и повышая рентабельность выполняемых им функций.

В основе современных математических методов поддержки принятия решений могут лежать нечеткая логика, нейронные сети для финансовых прогнозов, векторная оптимизация и формирования обобщенных показателей качества, экспертные оценки. Реализуемые в СППР алгоритмы и математические методы являются надежными помощниками при подготовке принятия решения, при этом аналитическое наполнение СППР зависит от решаемых задач, с которыми сталкивается аналитик в процессе обработки исходной информации. СППР способна играть роль опытного консультанта при подготовке к деловым переговорам, при стратегическом анализе рынка и составлении прогнозов в финансовой сфере. В условиях жесткой конкуренции СППР помогает руководству и ответственным сотрудникам фирм принимать обоснованные решения и может дать весьма квалифицированную оценку основных экономических параметров позволяющие взвешивать финансовые риски и принимать обоснованные решения.

Основными блоками СППР являются блоки анализа проблемы и принятия решений, база данных, база моделей, база знаний, интерфейс- "пользователь-система". Блоки анализа проблемы и принятия решений включают процедуру и методы, позволяющие с помощью баз данных, моделей и знаний сформулировать поставленную задачу, проанализировать возможность ее решения и получить результат. Интерфейс является интегрирующим компонентом и содержит

средства генерации и управления диалогом, управления данными, моделями, знаниями. Это один из наиболее существенных факторов всей концепции СППР. Высшее руководство предприятия зачастую видит только этот компонент информационной системы, поэтому успех СППР во многом связан не только с содержанием, но и с возможностями изобразительного ряда для представления результатов анализа и моделирования, будь то в электронной или бумажной форме.

Таким образом, цель разработки и внедрение системы поддержки принятия решений - информационная поддержка оперативных возможностей и создание комфортных условий для высшего руководства и ведущих специалистов при принятии обоснованных решений, соответствующих миссии предприятия, а также его стратегическим и тактическим целям. В этой связи широкие возможности для разработки информационной системы и пользовательского интерфейса предоставляет пакет MS Office. Встроенные возможности визуального программирования позволяют создавать специализированные приложения для обработки данных.

В основе прототипа СППР предлагается использовать методы имитационного моделирования и экспертных оценок, благодаря которым аналитик может учитывать влияние динамически изменяемых факторов окружающей среды на результирующие показатели инвестиционного проекта с последующей оценкой степени риска реализации данного проекта, а также учета мнений экспертов при принятии решений руководителем предприятия. В последующем предполагается реализация системы, способной на основе исходной информации генерировать отчеты полученных результатов в соответствии с методикой построения раздела "Финансовый план" бизнес-плана, в которую в качестве предоставляемых возможностей для анализа, будут интегрированы модули, реализующие методы имитационного моделирования и экспертных оценок.

Литература:

1. Идрисов А.Б., Картышев С.В., Постников А.В. Стратегическое планирование и анализ эффективности инвестиций. - М.: "Филинь", 1998. - 272 с.
2. Бизнес-план инвестиционного проекта. Практическое пособие / Под ред. Иванниковой И.А. - М.: "Экспертное бюро -М", 1997. 112 с.
3. Петровский А.Б., Шепелев Г.И. Система поддержки принятия решений для конкурсного отбора научных проектов // Проблемы и методы принятия уникальных и повторяющихся решений. - М.: ВНИИСИ, 1990. - С.25-31.

СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ФОНДОВОЙ БИРЖИ «ОМЕГА»

Железко Б.А., Морозов П.Е., Неправский А.А.
Белорусский государственный экономический университет
220672 г. Минск Партизанский пр. 22а БГЭУ ВШУБ

Целью создателей деловой игры «ОМЕГА» явилось создание инструмента для подготовки и переподготовки специалистов, владеющих основными биржевыми механизмами посредством имитационного игрового моделирования биржевой деятельности. При этом основной акцент в игре был сделан на использование преимуществ технологии имитационного игрового моделирования в ходе учебного процесса в вузах экономического профиля при подготовке и переподготовке специалистов, занятых в финансовой и банковской сферах экономики. Поэтому данная деловая игра может представлять интерес не только для вузов экономического профиля, но и кредитно-финансовых учреждений, банков, а также разработчиков деловых (тренинговых) игр.

Ключевые слова: деловая игра, фондовая биржа, моделирование биржевых процессов, подготовка и переподготовка специалистов.

Деловая игра «ОМЕГА» представляет собой систему имитационного моделирования фондовой биржи. Данный программный пакет, создан в среде Borland Delphi на основе адаптации и развития имитационной модели биржи «BOERSIMO», разработанной в 1994 г. в Университете Бундесвера в Гамбурге (Германия). Не секрет, что для успешной подготовки специалистов, владеющих основными биржевыми механизмами, необходимо скорейшее освоение ими основных принципов формирования и взаимодействия субъектов рынка. Использование при этом имитационного игрового моделирования биржевой деятельности позволяет группам слушателей проводить серии экспериментов при обучении использованию биржевых схем взаимодействия на базе современных информационных технологий. Тем самым значительно расширяется арсенал стратегий, которые они смогут применить в будущем в реальной работе. Эти обстоятельства и были положены авторами в основу при разработке данной деловой игры и позволяют использовать ОМЕГУ для подготовки и переподготовки кадров в финансовой и банковской сфере. Обратимся к сути деловой игры. Так, участниками ОМЕГИ являются студенты и преподаватель, которые могут покупать и продавать акции предприятий, котирующихся на бирже. При этом препода-

даватель выполняет роль биржевого маклера и регулирует весь процесс торгов. В деловой игре может быть задействовано до 400 участников, которые имеют возможность покупать и продавать акции максимум 20 акционерных обществ. Каждое акционерное общество могут представлять до 20-ти участников, при этом все они имеют правовую форму общества с ограниченной ответственностью. Как уже отмечалось выше, в деловой игре ОМЕГА используется имитационное игровое моделирование биржевой деятельности. Эта методика была взята за основу в связи с тем, что основной целью подготовки специалистов в данной области экономики является скорейшее освоение ими основных механизмов формирования и взаимодействия субъектов рынка. Основу деятельности таких механизмов, как показывает имеющийся мировой опыт эволюционного развития в направлении высокоорганизованной рыночной экономики, составляет эффективная реализация принципов предпринимательства и конкуренции. Компонентами практического воплощения этих принципов являются рыночные структуры (например, аукционы, биржи, торги) и специалисты, владеющие основными биржевыми механизмами. Причем знания этих механизмов и навыки принятия соответствующих решений необходимы каждому специалисту для нормальной адаптации к жизни в рыночных условиях. Проект деловой игры ОМЕГА разрабатывается при активном участии и поддержке:

- Высшей Школы Управления и Бизнеса при Белорусском Государственном Экономическом Университете (профессор Велеско Е.И.);
- Кафедры информационных технологий Белорусского Государственного Экономического Университета (профессор Морозевич А.Н.);
- Фонда Карла Дуйсберга (Кельн, Германия) (доктор Ханс Борман);
- Университета Бундесвера (Гамбург, Германия) (профессор Тидо Бетхер).

К областям, где возможно применение деловой игры ОМЕГА можно отнести подготовку и переподготовку кадров в финансовой и банковской сфере, решение ряда теоретических и практических задач (например, при выборе рациональных схем организации валютных аукционов, фондовых и товарно-сырьевых бирж, конкурсных торгов; моделирования коалиционного взаимодействия; комплексного контроля результатов обучения, создании программ-генераторов деловых игр и так далее). При этом существуют две основные возможности применения рассматриваемой деловой игры. Первая – это автономное использование, при котором в произвольно выбранном порядке производятся расчеты по поданным заявкам в одном из двух вариантов – по курсу макси-

мального оборота или с использованием метода ситуационного моделирования. При использовании метода ситуационного моделирования игроки ставятся руководителем игры в рамки определенной, зачастую реальной биржевой ситуации с использованием соответствующих этой ситуации реальных биржевых курсов. В этих условиях игрок должен определить ту оптимальную модель поведения на бирже, которая могла бы помочь ему наиболее эффективно использовать имеющиеся в его распоряжении финансовые ресурсы, рационально распорядиться имеющимся в его распоряжении пакетом ценных бумаг. В случае моделирования процесса расчета курса максимального оборота игрок имеет возможность увидеть, каким образом изменяется курс на основе воздействия участников на ситуацию на рынке. Участники на основе информации о курсах и оборотах акций, ситуации на рынке и в экономике в целом и некоторых других факторов должны принимать решения и подавать заявки на покупку и продажу акций. После каждых торгов на бирже для мотивации участников проводится ранжирование по двум критериям: 1) максимизация результатов деятельности; 2) успешное влияние на курсообразование. Далее выводится общая оценка, как средняя взвешенная двух рангов. Вторая возможность использования ОМЕГИ подразумевает ее использование совместно с системой имитационного моделирования управления предприятием ДЕЛЬТА. В этом случае ОМЕГА выступает, как дополнительный модуль ДЕЛЬТЫ, служащий для расширения масштабов моделирования деятельности предприятия на базе внедрения в имитационную модель основных биржевых механизмов, которые предприятия используют в реальной деятельности. Это весьма полезно как с точки зрения полноты анализа, так и глубины имитационной модели предприятия. Следует также немного осветить аспекты дальнейшего развития данного проекта. Так, в дальнейшем планируется совместное использование ее результатов с деловой игрой Дельта, которая является системой имитационного моделирования конкурирующих на рынке предприятий (например, для продажи на бирже акций разорившихся предприятий, смены собственника и так далее). Кроме того, это дальнейшее совершенствование графического интерфейса игры, полная или частичная реализация деловой игры, как системы искусственного интеллекта, например, для моделирования деятельности части участников (создание игроков-роботов), дополнение программы блоками обучения теоретическим знаниям, сбора статистики, подведения итогов игры и анализа стратегий поведения на бирже, моделирования деятельности других рыночных структур (например, банков, аукцио-

нов, типов моделируемых биржевых операций и так далее), а также погружение деловой игры в среду ИНТЕРНЕТ, имитация телекоммуникационного взаимодействия с одной из реальных бирж.

ИСПОЛЬЗОВАНИЕ IDEF0-МЕТОДОЛОГИИ ПРИ РАЗРАБОТКЕ БАЗ ДАННЫХ

Лавренова Н. М.

Белорусский государственный экономический университет

Управленческая внутрифирменная информационная система (УВИС) должна реализовывать совокупность информационных процессов удовлетворения потребностей в информации на разных уровнях принятия решений. Существует большое количество различных технологий разработки баз данных для УВИС. В данной работе рассматривается способ разработки модели базы данных корпоративной системы на основе функционального моделирования бизнес-процессов.

Ключевые слова: IDEF0-методология, бизнес-процессы, управленческая внутрифирменная информационная система (УВИС), информация, корпоративная система.

Для моделирования бизнес-процессов обычно используется методология SADT (точнее ее подмножество IDEF0). IDEF0-методология обеспечивает создание модели, которая является структурным изображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции [1]. IDEF0-модель состоит из диаграмм, фрагментов текста и глоссария, которые имеют ссылки друг на друга. Диаграммы - главные компоненты модели. На диаграммах все функции производственной системы и интерфейсы представлены как блоки (функции) и дуги (интерфейсы). Одной из наиболее важных особенностей методологии IDEF0 является постепенное внедрение все больших уровней детализации по мере создания диаграмм, отображающих модель. Таким образом, обеспечивается обозримое представление информации, и разработчик располагает хорошо очерченным предметом изучения с приемлемым объемом новой информации на каждой следующей диаграмме. Ниже приведен пример применения IDEF0-методологии для разработки базы данных для бухгалтерского учета. Пример представляет собой иллюстрированный вариант моделирования бизнес-процессов для демонстраций реализованного автором подхода. На первом этапе разработки строятся так называемые модели "Как

есть» и «Как будет». Модель «Как есть» разрабатывалась с помощью программного продукта фирмы Logic Works BPWin [2], поддерживающего методологию IDEF0. В ней отражена сущность процесса расчетов с подотчетными лицами и его учета. Процесс учета расчетов с подотчетными лицами состоит из 6 функциональных блоков:

1. подписать документы;
2. начислить деньги на командировочные расходы;
3. получить деньги в кассе;
4. съездить в командировку;
5. сдать авансовый отчет;
6. оформить бухгалтерские проводки.

Перед построением модели «Как будет» определялись в модели «Как есть» затраты на выполнение каждой функции во временном и денежном выражении. Эти данные представлены в таблице.

Таблица 1. Затраты на выполнение отдельных функций

Название функции	Затраты времени, ч	Сумма затрат, тыс.руб.
Подписать документы	1	97,79
Начислить деньги на командир. Расходы	4	391,14
Получить деньги в кассе	1	97,79
Съездить в командировку	1	97,79
Сдать авансовый отчет	1	97,79
Оформить бухгалтерские проводки	4	391,14
Всего:	11	1075,65

В результате анализа данных о затратах можно увидеть, что самые большие затраты времени и денег у функций «Начислить деньги на командировочные расходы» и «Оформить бухгалтерские проводки». Если выполнение данных функций вместо бухгалтера поручить автоматизированной системе, то данные изменятся (рис.1).

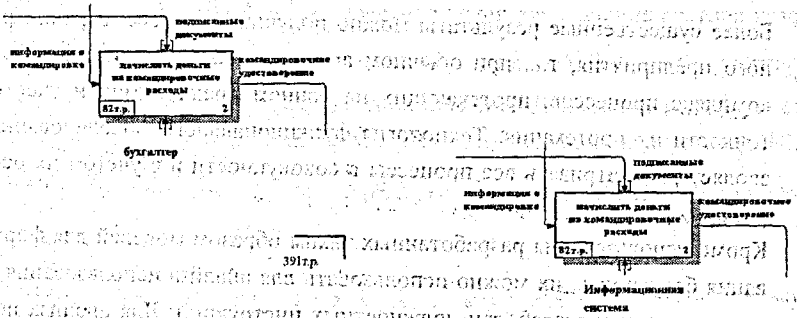


Рис.1. Функция «Начислить деньги на командир: расходы» в моделях «Как есть» и «Как будет»

Сравним качество информации в моделях «Как есть» и «Как будет»:

- 1. По целенаправленности:**
 В модели «Как есть» о целенаправленности информации говорить очень сложно, тогда как при сетевой базе данных в модели «Как будет» возможно быстрое получение информации для удовлетворения конкретных потребностей.
- 2. По объекту и качеству:**
 В модели «Как есть» информация поступает в том виде, в котором она хранится, т. е. она зачастую избыточна; имеет большой объем, формулировки могут быть не четкими применительно к ситуации, в которой информация понадобилась. В модели «Как будет» информация может быть представлена незамедлительно и в нужном объеме, т. к. при автоматизированном хранении информации время для выполнения запроса по ней достаточно мало.
- 3. По точности и достоверности:**
 Систематизация в модели «Как будет» более оптимальная по сравнению с моделью «Как есть».
 Построение схемы базы данных произведем с помощью программного продукта ERWin на основе модели «Как будет».
 В процессе анализа полученных результатов выявлено, что:
- 4. В результате получения новой модели стоимость процесса уменьшилась с 1075,65 до 440,05 тыс. руб., т.е. в 2,4 раза. Поэтому можно сделать вывод, что автоматизирование данных функций целесообразно.**
- 5. Сравнительный анализ качества информации в моделях «Как есть» и «Как будет» показал ее значительное улучшение по рассмотренным параметрам.**

Более существенные результаты можно получить при рассмотрении реального предприятия, т.к. при обычном анализе сложно порой охватить весь комплекс процессов, протекающих на данном предприятии, и учесть все тонкости их протекания. Технология функционального моделирования позволяет рассматривать все процессы в совокупности и с учетом их особенностей.

Кроме использования разработанных таким образом моделей для формирования баз данных, их можно использовать для анализа использования рабочего времени и разработки должностных инструкций. Для анализа использования рабочего времени для функций определяются временные характеристики, на основе которых и производится собственно сам анализ.

Для разработки должностных инструкций на основе разработанной модели определяется перечень работ, выполняемых данным должностным лицом, а также определяется степень доступа к информации.

Из всего выше сказанного можно сделать вывод, что использование IDEFO-методологии для анализа и реинженеринга бизнес-процессов является достаточно перспективной областью научных исследований.

Литература:

1. Железко Б.А., Морозевич А.Н. Информационно-аналитические системы поддержки принятия решений. — Мн.: НИУ, 1999 г. — 140 с.
2. Маклаков С.В. BPWin и ErWin — Case-средства разработки информационных систем. — Москва: «Диалог-мифи», 1999 г. — 256 с.

ОЦЕНИВАНИЕ НАЧАЛЬНЫХ ЗНАЧЕНИЙ АВТОРЕГРЕССИОННЫХ ВРЕМЕННЫХ РЯДОВ

Зеневич Д.В.

Белорусский государственный университет, Минск, пр. Ф. Скорины, 4

Аннотация. В работе рассмотрен новый метод статистического оценивания начальных значений временных рядов, описываемых моделью авторегрессии.

Ключевые слова: авторегрессия, начальные значения, оценивание, МНК.
Пусть наблюдается временной ряд $\{x_t\}$, описываемый моделью авторегрессии порядка p : $(AP(p))$:
$$x_t = \theta^0 X_{t-1} + \xi_t, \quad t = 1, 2, \dots, T, \quad (1)$$

где $X_{t-1} = (x_{t-1}, \dots, x_{t-p})'$, $\theta^0 \in \mathbf{R}^p$ — неизвестное истинное значение вектора коэффициентов авторегрессии, $\{\xi_t\}$ — н.о.р.с.в., $L\{\xi_t\} = N(0, \sigma^2)$, T — длительность наблюдения, “ ’ ” — знак транспонирования. Для оценивания параметров модели (1) чаще всего используют метод наименьших квадратов (МНК) [1], [2]: $\hat{\theta} = \arg \min_{\theta} F(\theta, X_0)$, где $F(\theta, X_0) = \sum_{t=1}^T (x_t - \theta X_{t-1})^2$. Явный вид этой оценки:

$$\hat{\theta} = A^{-1}a, \quad \text{где } A = \sum_{t=1}^T X_{t-1}X_{t-1}', \quad a = \sum_{t=1}^T x_t X_{t-1}' \quad (2)$$

Легко видеть, что вектор a и матрица A зависят от вектора начальных значений $X_0 = (x_0, x_{-1}, \dots, x_{1-p})'$, то есть $\hat{\theta} = \hat{\theta}(X_0)$. При теоретических исследованиях вектор X_0 предполагается заданным, хотя на практике он обычно неизвестен. Таким образом, задача оценивания параметров авторегрессии приводит к необходимости оценивания ее начальных значений.

Наиболее известны следующие методы оценивания начальных значений [3]:

1) изменить временные индексы: $x_{t-p} := x_t$, $t = \overline{1, T}$; 2) положить $X_0 := E\{X_0\}$; 3) присвоить $X_0 := (0, \dots, 0) \in \mathbf{R}^p$; 4) построить “прогноз назад”. Обычно в литературе применяют один из этих методов без каких-либо обоснований. Чаще всего используется 1-й метод, но из-за “усечения” временного ряда на p наблюдений он не очень хорош для коротких временных рядов ($T < 4p$).

(1) МНК-оценка начальных значений

Для устранения произвола в выборе начальных значений авторегрессии предлагается новый метод, позволяющий идею получения МНК-оценок:

$$\hat{X}_0 = \arg \min_{X_0} F(\hat{\theta}, X_0). \quad (3)$$

Займемся поиском явного вида оценки (3). Определим $(p \times p)$ -матрицу сдвига S_p , у которой левый нижний $(p-k) \times (p-k)$ -блок является единичной матрицей, а остальные элементы нулевые. Введем обозначения:

$$L = \sum_{i=0}^{p-1} x_{t+i} S_i', \quad G = G(X_0) = \sum_{i=0}^{p-1} (S_i X_0 X_0' S_i' + S_{p-i} X_p X_p' S_{p-i}' + S_i X_0 X_p' S_{p-i}'),$$

$$h = \sum_{i=0}^{p-1} x_{t+i} S_{p-i}' X_p + \sum_{i=p}^{T-1} x_{t+i} X_i, \quad K = \sum_{i=0}^{p-1} S_{p-i}' X_p X_p' S_{p-i}' + \sum_{i=p}^{T-1} X_i X_i'$$

Введем в рассмотрение любую матричную норму $\|\cdot\|_m$ и любую векторную норму $\|\cdot\|_m$, согласованные между собой. Определим величину $w = \|K\|_m^{-1} \|h\|_m^2$.

Теорема 1. Для МНК-оценки вектора начальных значений X_0 справедливо

$$\hat{X}_0 = Z^{-1}z + O(w)I_p \quad (4)$$

где I_p — матрица единиц, $z = LK^{-1}h - \sum_{i=0}^{p-1} H'K^{-1}S'_{p-i}X_p S'_i K^{-1}h$,

$$Z = \sum_{i=0}^{p-1} (S'_i K^{-1} h h' K^{-1} S_i + H' K^{-1} S'_{p-i} X_p S'_i K^{-1} L + L' K^{-1} S_i X_p S'_{p-i} K^{-1} h + S'_i K^{-1} h X'_p S'_{p-i} K^{-1} L + L' K^{-1} S'_{p-i} X_p H' K^{-1} S_i) - L' K^{-1} L.$$

Используя определение количества информации по Шеннону можно показать, что для гауссовского стационарного временного ряда с некоторого момента t_0 все последующие наблюдения не дают значимой информации об X_0 . Это доказывает невозможность построения состоятельной оценки вектора X_0 по одной реализации $\{x_t\}$, $t = \overline{1, T}$.

Обобщение модели: случай M реализаций

Рассмотрим следующую обобщенную модель. Пусть наблюдается M независимых реализаций процесса, описываемого моделью $AP(p)$ с фиксированным вектором начальных значений X_0 , т.е.

$$x_t^{(m)} = \theta^m X_{t-1}^{(m)} + \xi_t^{(m)}, \quad X_0^{(m)} = X_0, \quad m = \overline{1, M}; \quad t = \overline{1, T_m}$$

где M — номер реализации, каждая реализация удовлетворяет (1), ошибки независимы как внутри одного процесса, так и между ними:

$$E\{\xi_t^{(i)} \xi_s^{(j)}\} = \sigma^2 \delta_{ij} \delta_{ts}, \quad \delta_{ij} - \text{символ Кронекера. Таким образом, неизвестны } 2p + 1$$

параметров: $\theta^0, X_0 \in \mathbf{R}^p, \sigma^2$. Воспользовавшись методом максимального

правдоподобия, используя обозначение $F_M = \sum_{m=1}^M \sum_{t=1}^{T_m} (x_t^{(m)} - \theta X_{t-1}^{(m)})^2$,

получаем оценки:

$$\hat{\theta} = \arg \min_{\theta} F_M(\theta, X_0), \quad \hat{X}_0 = \arg \min_{X_0} F_M(\hat{\theta}, X_0), \quad s^2 = F_M(\hat{\theta}, X_0) / \sum_{m=1}^M T_m - 1. \quad (5)$$

Явный вид оценок параметров такой расширенной модели строится аналогично случаю одной реализации. Добавим индекс (m) ко всем ранее полученным результатам для обозначения принадлежности результатов m -й выборке. На-

пример, обозначим $A^{(m)} = \sum_{i=1}^T X_{i-1}^{(m)} X_{i-1}^{(m)'}$, $a^{(m)} = \sum_{i=1}^T x_i^{(m)} X_{i-1}^{(m)'}$ для формулы (2). Тогда для обобщенной модели все предыдущие результаты справедливы после добавления суммирования по m . Представим здесь только два результата.

$$\hat{\theta} = A_M^{-1} a_M, \quad \text{где } A_M = \sum_{m=1}^M A^{(m)}, \quad a_M = \sum_{m=1}^M a^{(m)}$$

Теорема 2. Пусть наблюдается M реализаций процесса авторегрессии с общим вектором начальных значений X_0 . Тогда МНК-оценка (3) вектора X_0 с точностью до $O(w_M)$ имеет вид $\hat{X}_0 = Z_M^{-1} z_M + O(w_M) I_p$,

где $Z_M = \sum_{m=1}^M Z^{(m)}$, $z_M = \sum_{m=1}^M z^{(m)}$, $w_M = \sum_{m=1}^M w^{(m)}$, а величины $Z^{(m)}$, $z^{(m)}$, $w^{(m)}$ определяются согласно теореме 1.

Численные эксперименты

Результаты теорем 1, 2 проверялись во время многочисленных компьютерных экспериментов. При этом генерировалось N случайных авторегрессионных временных рядов, после этого вычислялось отклонение $\Delta_i = \|\hat{X}_0^{(i)} - X_0\|^2$, где $\hat{X}_0^{(i)}$ — оценка вектора X_0 в i -м эксперименте, и проводилось усреднение $\Delta = \sum_{i=1}^N \Delta_i$. Ниже представлены результаты экспериментов с моделью $AR(2)$ при $\theta^0 = (1.1447, -0.3237)'$, $X_0 = (6.6903, 4.1429)'$, $\sigma^2 = 63.6982$, $T_m = 20$, $t = 1, M$.

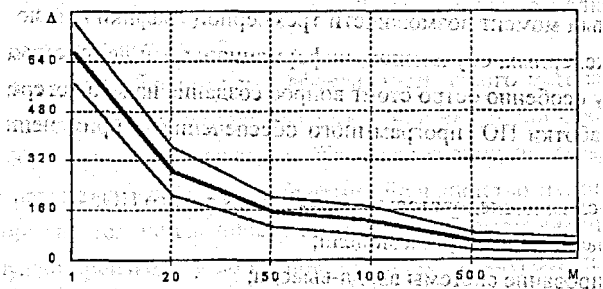


Рис. Зависимость $\Delta(M)$ с 95% доверительными интервалами.

Результаты экспериментов показывают применимость предложенного метода оценивания начальных значений в практических приложениях.

ЛИТЕРАТУРА

1. Айвазян С.А., Енюков И.С. Прикладная статистика. — М., 1983.

2. Андерсон Т. Статистический анализ временных рядов. — М., 1976.
3. Бокс Дж., Дженкинс Г. Анализ временных рядов. — М., 1974.
4. Kharin Yu.S., Zenevich D.V. *Robastness of Autoregressive Forecasting under Misspecified Model*. Computer Data Analysis & Modeling. Proceedings of the V-th Int. Conf. Minsk, BSU, 1998, pp: 120–126.

КОНСОЛЬНОЕ УПРАВЛЕНИЕ РЕСУРСАМИ В ЗАДАЧАХ ПОСТРОЕНИЯ ТРЕХМЕРНОГО ИЗОБРАЖЕНИЯ

Бахтин В.В., Круглов Д.Г.

Белорусский государственный университет информатики и радиоэлектроники
г.Минск, 220027, ул.П. Бровки 6

Аннотация: В данной статье рассматривается задача повышения качества библиотек визуализации трехмерной графики в реальном масштабе времени. Приводится метод ускорения и упрощения этапа тестирования и отладки графической модели, основанный на консольном управлении.

Ключевые слова: Библиотеки визуализации трехмерной графики, отладка и тестирование трехмерной модели, объектно-ориентированное построение библиотеки.

В последние годы системы визуализации трехмерной графики получили небывалое развитие. Бурный рост производительности персональных компьютеров сделал трехмерное моделирование доступным широкому кругу пользователей. В данный момент возможности трехмерной графики во всю используются в ряде инженерных, обучающих, информационных и демонстрационных систем. По этому особенно остро стоит вопрос создания надежных средств автоматизации разработки ПО (программного обеспечения) с применением трехмерной графики.

Перечислим последовательность создания такого ПО:

- проектирование трехмерной модели;
- программирование системы ввода-вывода;
- создание трехмерных графических ресурсов;
- тестирование и отладка модели;
- сдача в эксплуатацию.

В рассмотренной последовательности остановимся подробнее на этапе тестирования и отладки модели. Завершающая стадия работы над трехмерной моделью — одна из самых сложных. Связанно это с тем, что система ввода-

вывода и физическая модель уже запрограммированы. В такой ситуации очень тяжело находить и исправлять ошибки в ресурсах, или что еще хуже – в общей концепции построения модели. Необходимость возвращаться к этапу создания ресурсов для внесения самых мелких корректив, необходимость неоднократной компиляции (для больших по объему проектов это имеет особое значение) при обнаружении мелких ошибок отнимает массу времени. Тем более что с ресурсами чаще всего работает не программист, а дизайнер. В результате на отладку модели уходит большой объем времени.

Этот этап можно упростить, используя консольное управление. Анализ крупнейших мировых систем трехмерной визуализации показал эффективность применения такого подхода. Так, например, известные библиотеки LightTech, Geometry Vox II, а также Quake Engine содержат в себе консоль.

Поясним подробнее, что означает термин «консольное управление» в рамках графической трехмерной системы. Из названия (консоль) становится понятно, что это альтернативный интерфейс. Однако это интерфейс не с пользователем, а с разработчиком. Интерфейс на более низком уровне. Чаще всего он ограничен одной строкой ввода, и одной строкой вывода (возможна область вывода). Консольные команды предоставляют доступ ко всем элементам модели, к их свойствам и методам. Более того, они позволяют изменять параметры системы вывода; «на лету» менять такие настройки, как разрешение, цветность, наличие освещения, текстурирование и пр. Таким образом, консоль является идеальным инструментом отладки модели. И выполняется отладка в процессе работы модели, а не отдельно.

Рассмотрим подробнее, в чем эффективность такого метода. Множество мелких ошибок модели появляются вследствие неточного или невнимательного размещения объектов в пределах сцены. Кроме того ошибки связывания объектов, ошибки параметров поведения объектов. Большинство из них достаточно очевидно исправляются, но сложность состоит в требуемом для этого времени. Изменения нужно производить во внешнем редакторе ресурсов, или, что сложнее, в коде программы с последующей компиляцией. Консоль позволяет исправить ошибку (изменить состояние модели) прямо в процессе работы. Проверить полученный результат и продолжить работу. По мере накопления информации об исправлениях, можно произвести общее исправление в ресурсах и коде. Причем, за счет предварительной проверки, можно быть уверенным в том, что эти исправления не повлекут нарушения работы системы.

В разрабатываемой нами трехмерной графической системе визуализации, консоль реализована в большем объеме, чем в приведенных выше мировых аналогах. Фактически, при расширении возможностей консоль превращается в мини-транслятор, обрабатывающий команды языка управления ресурсами. В нашем случае, вся библиотека визуализации строится как иерархия классов (метод ООП). Каждый класс, за счет наследования, получает ряд функций по обработке запросов от консоли. Это могут быть вызовы методов класса, получение или установка значения переменных класса. В процессе работы все объекты системы получают уникальные имена и доступны по этим именам. Консоль, после поступления и интерпретации команды, находит нужный объект и передает ему интерпретированную команду. Весь процесс выглядит следующим образом:

- поступление команды;
- ее интерпретация консолью;
- определение необходимых параметров;
- нахождение объекта по его имени;
- передача объекту команды;
- передача объекту параметров;
- выполнение объектом команды.

В интерпретацию команды входит определение типа команды (вызов функции, создание/удаление объекта, получение значения переменной и т.д.). Далее, определение имени объекта и передаваемых ему параметров.

Нахождение любого объекта по имени является задачей отдельного системного класса. Отметим лишь, что это простейшая задача работы со списками. При передаче объекту команды, консоль «не знает» подробностей обработки команды. Каждый объект может выполнять свои специфические команды, и он сам проверит правильность всех параметров. Команды совпадают с публичными (public) функциями и переменными класса, таким образом, разработчику нет необходимости изучать новый для него язык консольных команд. Достаточно знать имена объектов модели (они доступны в ресурсах) и функции этих объектов. Фактически, такая консоль является одним из методов создания транслятора C/C++ в пределах конкретной задачи.

Подобный подход трансляции консольных команд реализован в трехмерном редакторе 3DStudioMax. Однако там он расширен до полного языка программирования, и на его создание было потрачено значительный объем временных и людских ресурсов. Приведенный выше вариант консоли отличается тем,

что его реализация не отнимает дополнительного времени у разработчика, так как исполнение команд в объектах — это простое перенаправление параметров на уже описанные в объекте методы. А программирование консоли требует незначительных усилий.

В дальнейшем, лишь незначительно изменив условия поступления команд, можно обрабатывать командные файлы. Таким образом, возможна автоматическая инициализация системы, что значительно проще программной, так как позволяет изменять параметры после компиляции программы. Кроме того, возможно создание тестовых последовательностей, обработка которых каждый раз при старте системы гарантирует ее работоспособность. Реализация этих дополнений значительно упрощит работу разработчиков программных модулей многократного использования.

Реализовать консольное управление можно не только в графических системах визуализации. В большинстве систем, связанных с обработкой сложных моделей, хранящихся во внешних ресурсах может возникнуть необходимость приостановить работу системы, изменить параметры системы или модели и запустить ее опять. По мере накопления информации об ошибках и неточностях, а также точных методах их исправления производится перекомпиляция. Такой подход сокращает временные издержки на отладку любой системы.

ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Карасик Е. А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, ул. П. Бровки, 6

Аннотация: обзорная статья по основным методам и стратегиям тестирования программного обеспечения.

Ключевые слова: тестирование; программное обеспечение;

Одними из основных методов повышения надежности ПО, являются тестирование, отладка и верификация, которые состоят в проверке КП на соответствие заданной спецификации [1].

Тестирование программного обеспечения охватывает целый ряд видов деятельности. Сюда входят постановка задачи для теста, проектирование, написание тестов, тестирование тестов и, наконец, выполнение тестов, и изучение результатов тестирования. Решающую роль играет проектирование тестов. Существует целый спектр подходов к проектированию тестов [2].

При тестировании ПО, используя подход функционального тестирования, программу рассматривают как «черный ящик» (ее исходный текст не используется). Происходит проверка соответствия поведения программы ее спецификации. Критерием полноты тестирования при таком подходе является полный перебор всех возможных значений входных данных, (исчерпывающее тестирование), что является невыполнимой задачей.

При структурном тестировании программа рассматривается как «белый ящик» (т.е. ее текст открыт для пользования). Происходит проверка логики программы. Полным тестированием в этом случае будет такое, которое приведет к перебору всех возможных путей на графе передач управления программы (ее управляющем графе). Даже для средних по сложности программ числом таких путей может достигать десятков тысяч. Поэтому при структурном тестировании необходимо использовать другие критерии его полноты, позволяющие достаточно просто контролировать их выполнение, но не дающие гарантии полной проверки логики программы.

Но даже если предположить, что удалось достичь полного структурного тестирования некоторой программы, в ней, тем не менее, могут содержаться ошибки, т.к.

1. программа может не соответствовать своей внешней спецификации, что в частности, может привести к тому, что в ее управляющем графе окажутся пропущенными некоторые необходимые пути;
2. не будут обнаружены ошибки, появление которых зависит от обрабатываемых данных (т.е. на одних исходных данных программа работает правильно, а на других – с ошибкой).

Таким образом, ни структурное, ни функциональное тестирование не может быть исчерпывающим. Поэтому при построении тестов необходимо различные комбинации этих двух подходов.

Также следует отметить метод символического тестирования [3]. Этот метод позволяет проводить тестирование без конкретных исходных данных, т.е. использовать такой язык программирования, где вместо реальных данных фигурируют произвольные символы. Такое символическое выполнение является естественным расширением нормального (числового) выполнения. В этом случае в качестве результатов выдаются формулы с символическими значениями (т.е. арифметические формулы заменяются алгебраическими). Существуют два вида символического выполнения программ:

1. по заданному пути программы необходимо определить, каким условиям должны удовлетворять входные значения, чтобы при выполнении программы на этих данных управление передавалось бы именно по этому пути. Символическое выполнение по заданному пути программы можно осуществлять как с начала, так и с конца.
2. по зафиксированным ограничениям на символические входные данные определить путь (или пути) программы, который будет выполнен на этих данных, и символические значения переменных в конце пути.

Применение символического выполнения имеет следующие основные достоинства:

- а) символическое выполнение первого вида, которое применяется для генерации тестов, позволяет образовать тесты, учитывая логическую структуру программы. В этом случае удастся всесторонне проверить программу, например, автоматически составить тесты так, чтобы на них были выполнены все ветви программы;
- б) символическое выполнение второго вида дает возможность одним выполнением программы на символических входных данных заменить весьма большое число прогонов на конкретных значениях. Если при этом удастся разбить область входных значений программы на классы, где каждый класс содержит конкретные значения, эквивалентные с точки зрения логики программы, то выполнение программы на символических значениях, представляющих эти классы, довольно часто оказывается полным (и конечным) перебором всех возможных входных значений.

Вторым по важности аспектом тестирования (после проектирования тестов) является последовательность слияния всех модулей в систему или программу [2,3].

Основными стратегиями тестирования являются восходящее и нисходящее тестирование. Но применение этих двух стратегий в "чистом" виде обычно не применяется. Существует большой выбор различных подходов, которые могут быть использованы для слияния модулей в более крупные единицы, более подробно это описано в работе [2].

Из-за ограниченности ресурсов тестирования используются, прежде всего, для обнаружения наиболее опасных ошибок в наиболее важных режимах функционирования программ. С этой целью последовательно применяются методы

тестирования: статический, детерминированный, стохастический и в реальном масштабе времени.

Статическое тестирование является наиболее формализованным и автоматизируемым методом проверки корректности программ. В качестве эталона применяются правила структурного построения программных модулей и обработки данных; конкретизированные для проекта в целом. Проверка степени выполнения этих правил проводится без исполнения объектного кода программы путем формального анализа текста программы на языке программирования.

При детерминированном тестировании контролируется каждая комбинация исходных эталонных данных и соответствующая ей комбинация результатов функционирования программы. Это позволяет выявлять отклонение результатов от эталона с конкретным фиксированием все значений исходных и результирующих данных, при которых это отклонение обнаружено.

В сложных программах невозможно перебрать все комбинации исходных данных и проконтролировать результаты функционирования программы на каждой из них. В таких случаях применяется стохастическое тестирование, при котором исходные данные задаются множествами случайных величин с соответствующими распределениями и для сравнения полученных результатов используются распределения случайных величин.

В процессе тестирования в реальном масштабе времени проверяется исполнение программ, и обработка исходных данных с учетом времени их поступления, длительности и приоритетности обработки, динамики использования памяти и взаимодействия с другими программами и т.д.

Литература

1. Липаев В.В. Тестирование программ. — М.: Радио и связь, 1986.
2. Майерс Г. Надежность программного обеспечения. — М.: Мир, 1980.
3. Бахгизин В. В., Иблуду К. А., Савкин В. В. Методы тестирования и верификации программ. — М.: Машиностроение, 1984.

СИСТЕМА КОНТРОЛЯ ЗНАНИЙ

Тиунчик Д.В., Лобов С.Д.

Брестский государственный университет им. А.С.Пушкина

Брест, Беларусь

Аннотация: в работе рассмотрено построение пакета контролирующих программ в среде Windows 9x. Структура и особенности пакета, его возможно-

сти: создание тестов, проведение тестирования и анализ результатов тестирования.

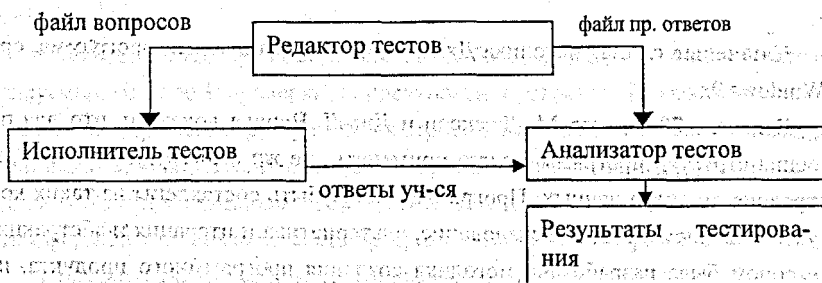
Ключевые слова: методика Джексона, контролирующая программа, среда Windows 9x.

В начале 70-х годов М. Джексон и Дж.-Д. Варнье доказали, что для построения структур программ можно применять те же конструкции, что и для построения структур данных. Программы могут быть составлены из таких конструктивных элементов, как следование, альтернатива и итерация и абстракция. Джексоном была разработана методика создания программного продукта, названная в последствии его именем. Эта методика основывается исключительно на процедурном подходе к программированию. Она позволяет максимально автоматизировать процесс программирования и свести к минимуму количество ошибок возникающих в процессе создания комплексов (пакетов) программного обеспечения.

Мы на практике применили методику Джексона при написании пакета тестирующих программ в одной из самых распространенных операционных средах – Windows 9x. При написании использовалась среда визуального программирования – Delphi.

Целью работы было создание оболочки для осуществления текущего и итогового контроля знаний тестируемого по самым различным учебным дисциплинам, а так же компьютерная реализация тестов из различных областей исследования (психология, социология, определение IQ и т.п.). Мы исходили из того, что оболочка должна отвечать следующим требованиям:

- удобный многооконный интерфейс;
- простота в использовании;
- доступная справочная система;
- поддержка различных типов вопросов, как с рисунками, так и без них;
- отсутствие доступа у тестируемого к ответам;
- наличие временных ограничений;
- автономные режимы создания тестов и анализа результатов тестирования.



Структура программы: пакет состоит из трех компонент: редактор тестов, исполнитель тестов и анализатор тестов.

Редактор тестов представляет собой многооконную среду для создания и редактирования тестов. Отметим основные особенности редактора тестов. С помощью редактора тестов создаются два файла: файл вопросов и файл правильных ответов. Файл вопросов имеет формат Paradox. В этом файле хранятся формулировки вопросов, рисунки к вопросам (если в них есть необходимость) и, если это необходимо, варианты ответов для выбора тестируемым. Файл правильных ответов редактор тестов создает автоматически. Файл правильных ответов — это также файл формата Paradox, в котором хранятся правильные ответы на вопросы теста из соответствующего файла вопросов. Такой способ подготовки тестов позволяет принять дополнительные меры по защите правильных ответов к тесту от несанкционированного доступа. Причем имеется возможность защиты данных паролем.

Исполнитель тестов служит для выполнения тестирования учащихся. При работе использует файл вопросов, созданный редактором тестов. Программа позволяет устанавливать различные параметры тестирования. Так с легкостью можно ограничить время тестирования, задать пароль для прерывания тестирования. Отметим также, что в режиме тестирования отключается ряд комбинаций "горячих" клавиш, в том числе и комбинация клавиш Ctrl+Alt+Del с целью предотвращения принудительного завершения работы исполнителя тестов. Программа позволяет проводить потоковое тестирование, в этом случае достаточно один раз указать файл вопросов и параметры тестирования в начальном диалоге и после этого «менять» тестируемых. По результатам тестирования программа автоматически создает файлы с ответами тестируемых. Эти файлы используют-

ся в последствии для анализа результатов тестирования с помощью анализатора тестов.

Анализатор тестов представляет собой программы анализа результатов тестирования. Требуется наличия файла правильных ответов и указания папки с ответами учащихся. Автоматически выполняет поиск файлов ответов учащихся и сравнение их с правильными ответами. В случае полного совпадения ответов выставляется «+», несовпадения ответов – «-». Если же ученик не смог ответить на поставленный перед ним вопрос, то выставляется знак «н/о» – «нет ответа», а в случае неполного, или избыточного ответа – «ч/о» – «частичный ответ». Все данные представлены в виде таблицы, в которой отображены индивидуальные данные о тестируемом, результаты ответа на каждый вопрос, а также общие результаты: количество правильных, неправильных и частичных ответов, а также количество вопросов, на которые ответа не последовало. Собирается также информация о суммарных результатах тестируемых, которая позволит проанализировать общие результаты тестирования всей группы сразу. Это позволит тестирующему проанализировать результаты, выяснив, например, степень усвоения материала по той или иной теме. Имеется возможность распечатки отчета о тестировании на любом принтере, установленном в системе.

Контекстно-зависимая справочная система приходит на помощь в любой ситуации. Поэтому любой неподготовленный пользователь программы успешно справляется с возникающими в процессе эксплуатации трудностями.

Далее поговорим о поддерживаемых параметрах вопросов. Как было отмечено выше, редактор тестов позволяет создавать, вопросы различных типов. Это, во-первых, вопросы альтернативного выбора, т.е. такие вопросы, при ответе на которые ученик выбирает правильный ответ или правильные ответы из нескольких предложенных ему альтернатив (от 2 до 5 вариантов ответа на вопрос). Во-вторых, редактор тестов с легкостью создает, такие вопросы, правильный ответ на которые необходимо вводить с клавиатуры. Для этого достаточно задать только один вариант ответа в редакторе тестов. В результате этого при выполнении теста будет отображена строка ввода, где и производится ввод правильного ответа с клавиатуры.

Литература

1. Кинг Д. Создание эффективного программного обеспечения: пер. с англ. - М.: Мир, 1991. - 228 с.

2. Тиунчик Д.В., Лобов С.Д. "Использование средств языка программирования DELPHI для написания приложений ведения баз данных" / Материалы Брестской региональной научно-практической конференции молодых ученых "Математика и ЭВМ-98", Брест, 1998. — С.44.

КОНСТРУКТИВНАЯ КОМПОЗИЦИЯ СТРУКТУР СЕТЕВЫХ МОДЕЛЕЙ ОРГАНИЗАЦИОННО-ТЕХНОЛОГИЧЕСКИХ СИСТЕМ

Бортник Е.М., Ревотюк М.П.

Белорусский государственный университет информатики и радиоэлектроники
Минск, ул. П.Бровки, 6, кафедра ИТАС

Рассматривается задача формального конструирования исполнительных версий сетевых интерпретируемых моделей в системах управления реального времени. Предлагаемый реляционный подход к структуризации моделей дискретных процессов оценивается как средство сокращения трудоемкости формализации и конструктивная основа объектно-ориентированного проектирования систем управления организационно-технологического уровня.

Ключевые слова: имитационное моделирование, сети Петри, реляционные операции.

Известно, что наиболее привлекательным видом моделирования дискретных процессов с регулярной структурой являются сети Петри и их расширения [1]. Ориентация сетей Петри на отражение свойства восприимчивости реальных систем к локальным изменениям переменных состояния весьма удобна как при формализации дискретных процессов со сложными асинхронными взаимодействиями, так и реализации технологий объектно-ориентированного проектирования и программирования [2]. Цель исследования — разработка формального аппарата структурных преобразований имитационных моделей на основе расширенных сетей Петри применительно к задачам управления системами организационно-технологического уровня.

Сущность задачи конструирования структурированных сетевых описаний поясним на примере моделирования робототехнической системы (РТС) с линейным размещением оборудования. Пусть моделируемая РТС, предназначенная для обработки транспортных партий деталей (ТПД), включает комплект обрабатывающего и накопительного оборудования и транспортный робот (ТР).

Предварительный анализ позволяет выделить в процессе функционирования РТС три качественно различные составляющие: прохождение ТПД по тех-

нологическому маршруту, обработка ТПД на некоторой операции и транспортировка ТПД к месту выполнения очередной операции.

Для построения рекуррентного описания каждая выделяемая составляющая должна иметь хотя бы один общий элемент, интерпретируемый как ресурс. В рассматриваемом варианте РТС таким общим элементом является ТПД. Выделенные составляющие можно представить в виде сетевых моделей: модель процесса обработки ТПД на единице оборудования, модель процесса функционирования ТР и модель прохождения ТПД по соответствующему технологическому маршруту.

Получение полной модели РТС означает объединение сетевых описаний ее элементов. Принципиально такое объединение можно реализовать путем графического изображения всех фрагментов и последующего кодирования графа полной сети. Общеизвестный недостаток такого подхода — громоздкость описания и значительный объем ручной работы при его кодировании. Далее будет показан более эффективный способ создания полной сетевой модели РТС, основанный на ее реляционном представлении.

Предварительный анализ фрагментов сетевых моделей показывает, что их общее реляционное описание представимо тройкой отношений — декларации доменов модели (ДДМ), описания структуры смежности (ОСС) и описания элементов модели (ОЭМ).

Отношение ДДМ определяет словарь используемых отношений:

$$\text{ДДМ} := [I, Q, T],$$

где I — идентификатор; Q — кардинальное число; а T — тип элемента сетевой модели, например, переход, позиция, вершина графов транспортной сети или технологических маршрутов.

Отношение ДДМ по определению нормализовано. Каждый из доменов ассоциируется на содержательном уровне либо с одним из элементов РТС, либо с некоторой связью между элементами РТС.

Связь фрагментов сетевого описания определяется отношением описания структуры смежности:

$$\text{ОСС} := [X, Y, W].$$

Для определения ОЭМ в рассматриваемом варианте РТС естественно выделение следующих временных параметров, относящихся к одной единице обрабатывающего или накопительного оборудования:

ТВ1 — продолжительность обработки ТПД;

ТВ2 (ТВ3) — продолжительность перемещения ТР к соседней единице оборудования, размещенной справа (слева),

ТВ8 (ТВ9) — продолжительность погрузки (разгрузки) ТПД.

Нормализованное отношение ОЭМ имеет вид

$$\text{ОЭМ} = [\text{ТВ1}, \text{ТВ2}, \text{ТВ3}, \text{ТВ8}, \text{ТВ9}].$$

Таким образом, сетевое рекуррентное описание РТС можно представить в реляционном виде как совокупность формально интерпретируемых таблиц нормализованных отношений. Назовем такое описание виртуальным сетевым описанием (ВСО). Его последующее использование неизбежно требует "развертывания" в полную сетевую модель, называемую далее реальной сетевой моделью (РСМ). Статическое описание РСМ определяется отношением структуры смежности вершин $X(i): \{<Y(i, j), w(X(i), Y(i, j))>\}$, $Y(i, j) \in X(i)$ и векторной функцией задержки переходов сети $A(i) : D(i)$. Здесь $x'(\cdot)$ — вершины, смежные по выходу с вершиной x .

Формально связь ВСО и РСМ задается графами компоновки оборудования и технологических маршрутов. Вершины таких графов по содержательной интерпретации совпадают, а дуги отражают пространственно-целевые связи материальных потоков.

Для построения РСМ необходимо:

- образовать множества экземпляров элементов ВСО, соответствующих вершинам и позициям РСМ;
- связать элементы РСМ по схеме, представляемой отношением ОСС.

Алгоритм преобразования ВСО в РСМ имеет вид.

- Шаг 1. Вычисление уникальных идентификаторов элементов РСМ, используя данные отношения ДДМ.
- Шаг 2. Грамматический разбор фраз X и Y из кортежей отношения ОСС.
- Шаг 3. Проверка допустимости образования ассоциации между базовыми множествами X и Y .
- Шаг 4. Образование множеств элементарных ассоциаций $\langle \{x\}, \{y\}, W \rangle$ с исключением кортежей с пустыми значениями элементов x и/или y .
- Шаг 5. Преобразование описания РСМ в формат статического описания сетевой модели.

Эффективность применения ВСО можно определить сравнением размерностей ВСО и РСМ:

$$|\text{ВСО}| = \text{sum}\{U(i), i = 1, 2, \dots, |\text{ДДМ}|\},$$

$|PCM| = \text{sum} \{U(i) \text{ ДДМ.}Q(i), i = 1, 2, \dots, |ДДМ|\},$

$U(i) = (ДДМ.T(i) = a) + (ДДМ.T(i) = b), i = 1, 2, \dots, |ДДМ|$

(здесь предполагается, что **a** - переход, **b** - позиция сети).

Вывод о полезности применения реляционного подхода в конкретной задаче можно делать с учетом того, что изменение структуры моделируемой РТС не влечет необходимости изменения ВСО.

Литература

1. Питерсон Дж. Теория сетей Петри и моделирование систем: Пер с англ. - М.: Мир, 1984. - 284 с.
2. Буч Г. Объектно-ориентированное с примерами применения: Пер с англ. - М.: Конкорд, 1992. - 519 с.

ПОИСК И ПРЕДСТАВЛЕНИЕ ДОПУСТИМЫХ ТЕХНОЛОГИЧЕСКИХ МАРШРУТОВ В УСТАНОВКАХ КОМПАУНДИРОВАНИЯ

Ревотюк М.П., Тихомирова Е.В.

Белорусский государственный университет информатики и радиоэлектроники

Минск, ул. П.Бровки, 6, кафедра ИТАС

Предлагается модель процесса управления маршрутами транспортировки компонент в установках компаундирования жидких или газообразных продуктов, позволяющая учесть в реальном времени технологические ограничения и условия реализации маршрутов. Рассматриваются алгоритмы поиска допустимых маршрутов и их представления посредством нумерации деревьев на графах.

Ключевые слова: дискретные процессы, оптимизация на графах, управление в реальном времени.

Технологические установки компаундирования, например, масла на нефтеперерабатывающих заводах, включают систему продуктопроводов, емкости для хранения исходных ингредиентов и готового продукта, насосы, задвижки, смесители. Задание на получение определенного продукта в некоторой емкости - приемнике характеризуется списком емкостей - источников, а требуемый состав выходного продукта обеспечивается установкой режимов работы активных элементов установки - насосов и задвижек. Предмет обсуждения здесь - решение в реальном времени задач координации активных элементов в процессе реализации динамически порождаемого множества маршрутов.

Технологическая установка в контексте задач управления маршрутами может быть представлена многодольным ориентированным графом. Вершины графа соответствуют активным элементам оборудования, а дуги отражают имеющиеся связи элементов. Разветвляющиеся продуктопроводы для учета свойства текучести также приходится представлять отдельной вершиной. Каждому отрезку продуктопровода, по которому возможно движение в обоих направлениях, должны соответствовать две дуги противоположной ориентации. Доли графа отражают классификацию вершин.

Формально задача поиска допустимых технологических маршрутов в системах компаундирования может быть определена следующим образом:

— дано задание $T = \{S, D\}$, где S — вершина-источник, D — вершина-приемник; необходимо на графе технологической установки найти все альтернативные трассы реализации задания T .

Допустимые трассы реализации задания характеризуются:

- наличием на соответствующем ей пути на графе некоторой активной вершины, например, насоса;
- вершины S и D могут совпадать, что соответствует заданиям на перемешивание содержимого емкости;
- хотя каждая трасса может быть задана перечнем вершин пути на графе технологической установки, с позиции управления интерес представляет состояние лишь активных вершин.

Представление допустимой трассы, в отличие от пути на графе, фактически должно соответствовать дереву достижимости вершин, представляющих с учетом свойства текучести задвижки в закрытом состоянии (исключая источник и приемник). Поиск такого дерева естественно реализуется после построения пути между вершинами S и D . Таким образом, базовая процедура алгоритма решения обсуждаемой задачи — перечисление путей и циклов на графе.

Для поиска всех допустимых технологических маршрутов в системе компаундирования может быть использован алгоритм поиска пути на графе «сначала вглубь, а затем в ширину» с полным перебором возможных путей из вершины-источника S к вершине — приемнику D [1]. Однако, учитывая потребность определения условий реализации маршрута в терминах состояния вершин графа, можно предложить эффективную по памяти и быстродействию модификацию известных алгоритмов поиска в глубину [1, 2].

Статическое описание графа технологической установки представлено в виде объединения структур смежности прямого и инвертированного описания связей вершин [2]:

ndot - количество вершин сети;

narc - количество дуг сети;

pdot - индексы списков исходящих дуг, $|\text{pdot}| = \text{ndot}+1$;

parc - списки исходящих дуг, $|\text{parc}| = \text{narc}$;

bdot - индексы списков входящих дуг, $|\text{bdot}| = \text{ndot}+1$;

barc - списки входящих дуг, $|\text{barc}| = \text{narc}$;

Состояние процесса поиска решения отображается на множествах:

xmrk - вектор меток вершин сети, $|\text{xmrk}| = \text{ndot}$;

xdot - смещение индекса просмотренных дуг, $|\text{xdot}| = \text{ndot}$;

qdot - область очереди анализируемых вершин, $|\text{qdot}| = \text{ndot}$;

prec - массив предшествующих вершин дерева путей, $|\text{prec}| = \text{ndot}$.

Множество состояний отдельных вершин графа:

passive - пассивный элемент трассы;

actived - активный элемент трассы;

visited - просмотренная вершина сети;

storage - пропускаемая вершина трассы.

После построения очередного пути **P**, например, с помощью процедуры DFS [2] алгоритм представления трассы имеет вид:

1. Сброс пометок всех вершин графа.
2. Выделение активных вершин текущего пути: все вершины пути **P** получают метку **actived**.
3. Построение списков пассивных вершин для обеспечения реализуемости пути: для каждой вершины пути **P** построить дерево достижимости активных вершин графа (вершины пути следует рассматривать в обратном порядке). Алгоритм построения дерева достижимости активных вершин графа реализует волновую схему просмотра дуг графа без различения их ориентации.

Процедура **ДОСТИЖИМОСТЬ(i)** определяет условия протекания потока через активную вершину **i**:

— для всех вершин, смежных по входу либо выходу с текущей вершиной **i**, не являющихся активными, т.е. не имеющих метки **actived**:

- a) установить метку **passive**, означающую, например, закрытое состояние задвижки;

б) если вершина i представляет разветвляющийся отрезок продуктопровода или емкость, то для всех вершин j , смежных по входу либо выходу с текущей вершиной i , рекурсивно выполнить ДОСТИЖИМОСТЬ(j).

Литература

1. Свами М., Тхуласираман К. Графы, сети и алгоритмы. - М.: Мир, 1984 - 455с.
2. Гудман С., Хидетниemi С. Введение в анализ и разработку алгоритмов. - М.: Мир, 1981. - 368 с.

АНАЛИЗ ПОТЕНЦИАЛЬНОЙ ЭФФЕКТИВНОСТИ КООПЕРАТИВНЫХ СХЕМ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Тихомирова Е.В.
Белорусский государственный университет информатики и радиоэлектроники
Минск, ул. П.Бровки, 6, кафедра ИТАС

Объект рассмотрения – оценка потенциальной эффективности решения задач дискретной оптимизации на вычислительных сетях. Обсуждаются возможные схемы кооперации вычислительных ресурсов сети, необходимые и достаточные условия использования принципа накопления опыта, приводятся результаты аналитического и имитационного моделирования.

Ключевые слова: дискретная оптимизация, вычислительные сети.

Многие задачи дискретной оптимизации [1,2] можно характеризовать тройкой

$$Z = \langle V, P, S \rangle,$$

где V - множество вариантов объектов, подлежащих оценке по заданному критерию; P - процедура получения оценки качества для каждого варианта из множества V ; S - процедура реализации вычислительной схемы решения исходной задачи, определяющая порядок применения процедуры P к элементам V .

Известно, что построение в приемлемые сроки строгих и, вместе с тем, изящных в математическом отношении схем решения практических задач дискретной оптимизации в большинстве случаев не представляется возможным. Часто складывается ситуация, когда:

- имеется алгоритм и программа P решения задачи оценки отдельного варианта из множества V ;

- существует некоторый вариант схемы S , гарантирующий решение задачи Z (например, простой или направленный перебор);
- реализация схемы S возможна на множестве параллельно функционирующих вычислительных устройств (например, ЭВМ, объединенных в локальную вычислительную сеть).

В подобных ситуациях естественно попытаться реализовать идею "накопления опыта", возникающего при параллельном рассмотрении отдельных вариантов. В [1] выделены необходимые и достаточные условия построения кооперативной схемы, а также определена структура процедуры анализа подмножеств вариантов. Показано, что в первом приближении использование кооперативных схем влечет квадратичное в зависимости от количества ЭВМ сокращение времени решения задач выбора. Последнее справедливо относительно границ пессимистической оценки.

Целью проводимого исследования - уточнение эффективности кооперирования ЭВМ в более общих случаях организации вычислений.

Анализируемые варианты кооперирования ЭВМ:

- предварительное разделение множества вариантов V между ЭВМ и автономное накопление опыта каждой отдельной ЭВМ;
- предварительное разделение множества вариантов V между ЭВМ и коллективное накопление опыта всеми ЭВМ;
- варианты из множества V назначаются на первую освободившуюся ЭВМ, причем все ЭВМ используют коллективное накопление опыта.

Очевидно, что вычислительная трудоемкость анализа всех вариантов V в конечном счете зависит от порядка их рассмотрения, а точнее, от того, насколько рано выбираются "хорошие" варианты. Предполагая равномерное распределение значений вычислительной трудоемкости W анализа вариантов на интервале $[a; b]$, легко заметить, что наилучший случай последовательности анализа соответствует условию $W(i) < W(i+1)$, а наихудший - $W(i) > W(i+1)$, $i=1, 2, \dots, |V|$.

Соответственно, при кооперации m ЭВМ, $m > 0$, время анализа вариантов в случае их линейного упорядочения по значениям W характеризуется зависимостью

$$t_i = \begin{cases} \frac{bx}{m}; x \geq \frac{a}{2}; \\ \left((a-x) + \frac{2x-a}{2m} \right) \frac{b}{m}, x < \frac{a}{2}; \end{cases}$$

После усреднения по всем значениям x , характеризующих конкретный закон изменения W получим среднее время решения задач на m ЭВМ:

$$t_i = \begin{cases} \frac{a^2 b}{2} + \frac{a^2 b(m+1)}{8m} \\ \int_0^{\frac{a}{2}} \frac{bx}{m} dx, x \in \left[0, \frac{a}{2} \right] \\ \int_{\frac{a}{2}}^a \left((a-x) + \frac{2x-a}{2m} \right) \frac{b}{m} dx; x \in \left[\frac{a}{2}, a \right] \end{cases}$$

Последнее выражение не учитывает процесс обмена сообщениями о рекордных оценках и образование очередей. Ввиду трудности учета последних факторов аналитической моделью, были проведены имитационные эксперименты с моделями трех перечисленных выше способов организации кооперативных схем. Статистическая обработка результатов имитационных экспериментов позволила выявить следующие математические зависимости:

$$t_1(m) = 0,98 * 0,95^m$$

$$t_2(m) = \frac{1}{m^{0,85}}$$

$$t_3(m) = \frac{1}{m^{0,90}}$$

Исследование влияния задержек на передачу сообщений о рекордных оценках было проведено на двух видах вычислительной среды – одно-ранговой сети и сети с управляемой структурой, соответствующей схеме передачи рекордных оценок.

Выводы по результатам моделирования:

1. задержки на передачу сообщения при равном количестве ЭВМ у сети с управляемой структурой всегда меньше, чем, у одно-ранговой;
2. при одно-ранговой организации сети время задержек прямо пропорционально количеству кооперируемых ЭВМ;
3. при использовании сети с управляемой структурой для каждого количества кооперируемых ЭВМ можно найти оптимальное, число групп по критерию минимума задержек времени на передачу сообщений о рекордных оценках и нахождения в очереди.

Так, для 60 кооперируемых ЭВМ оптимальное число групп колеблется в интервале от 8 до 12, для 90 ЭВМ – 11-12, 180 ЭВМ – 22-25, 240 ЭВМ – 28-35.

Литература:

1. Ревотюк М.П. Кооперативные схемы алгоритмов решения задач выбора на распределенных вычислительных системах. - Мн.: МРТИ, 1990. – 16 с.
2. Романовский И.В. Алгоритмы решения экстремальных задач. - М.: Наука, 1977. - 352 с.

ПРИМЕНЕНИЕ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ОБРАЗОВ В СРЕДСТВАХ ЗАЩИТЫ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Бахтизин В. В., Крапивин В. А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, 220600, ул. Платонова, 39, кафедра ПОИТ

Аннотация. Предложенный алгоритм позволяет просто и надежно определять пользователя по его биометрическим характеристикам. Достоинствами предложенного алгоритма являются простота использования, надежность и невысокая стоимость по сравнению с традиционными средствами. Данный метод базируется на аппарате распознавания и классификации образов.

Ключевые слова. Защита от НСД, аутентификация, клавиатурный почерк.

В связи с распространением корпоративных сетей все острее встает проблема однозначной идентификации пользователя при входе в сеть. В связи с необходимостью работ одновременно с несколькими серверами, пользователь вынужден либо запоминать множество имен и паролей для получения доступа к каждому из нужных ему серверов, или применять дорогостоящие карточки.

Для решения этой проблемы используется метод автоматического выполнения запроса к серверу на получение доступа. При этом пользователь при первом доступе к серверу вводит пароль, а система на его рабочем месте запоминает его и при последующих сеансах работы подставляет этот пароль автоматически. Данный подход имеет ряд недостатков: пользователям необходимо наличие знаний об именах и паролях; для получения доступа к серверу достаточно провести атаку рабочего места пользователя, например при помощи программ «тройанский конь».

В связи с вышеизложенным предлагается следующий подход к защите от несанкционированного доступа. Пользователю или группе пользователей присваивается один пароль. Отпадает необходимость в запоминании какой-либо парольной информации конфиденциального характера, так как распознавание пользователя осуществляется не паролем, а с помощью математических методов.

Предлагаемый алгоритм защиты от несанкционированного доступа базируется на методах распознавания и классификации образов. При этом сам процесс определения пользователя использует его биометрические характеристики, а именно его клавиатурный почерк.

Собственно съем данных пользователя представляет собой следующую процедуру. Пользователь вводит требуемую фразу (то есть пароль), а программа фиксирует интервалы между нажатиями на соответствующие клавиши.

Рассмотрим некоторые методы распознавания образов, положенные в основу предложенного метода аутентификации. Более подробную информацию можно получить в [1, 2].

Пусть A - множество значений эталонов. При этом $A = \{x_i\}, i = \overline{1, N}$; N - объем выборки множества A . Множество определено на метрическом линейном пространстве размерности $k = s - 1$, где s - длина пароля в символах.

При этом выполняется условие $A = \bigcup_{j=1}^M A_j$, $j = \overline{1, M}$, где M - число пользователей в системе, и условие $A_i \cap A_j = \emptyset, \forall i, j = \overline{1, M}, i \neq j$. Данные пользователя, которые были с него сняты, это $x \in R^k$. В метрическом линейном пространстве задана метрика $d: R^k \times R^k \rightarrow R^1$. При этом возможно применение как l_p -метрики, так и менее известных метрик [1, 2]. Задача состоит в том, чтобы определить такой класс A_i , что $x \in A_i$.

В разработанном пакете программ используется несколько простых алгоритмов распознавания, в том числе метод k ближайших соседей и метод единственного эталона [1,2].

Пусть $F = \{f_s\}, s = \overline{1, S}$ - множество применяемых методов распознавания, где S - общее число методов. После того, как каждый метод выберет своего кандидата, проводится оценка полученных результатов. В данной системе используется метод коррекции распознавания по большинству [1,2].

Сформируем вектор $a = (a_1, a_2, \dots, a_M)$ так, что

$$a_i = \sum_{s=1}^S \mu_s (f_s(x) \in A_i), \forall i = \overline{1, M}, \quad (1)$$

где μ_s - вес соответствующего метода (обычно $\mu_s = 1$). Далее находим

$a_i = \max_{l=1, M} a_l$. Следовательно, $x_i \in A_l$. При этом, если

$$a_i < T \times \sum_{l=1}^M a_l \quad (2)$$

где T - порог прохождения кандидата в пользователи, то определение пользователя по введенным биоданным неоднозначно. Параметр $T \in [0; 1]$ задается вручную и показывает степень «строгости» проверки.

В процессе съема биоданных пользователь постепенно обучается, поэтому его параметры постепенно улучшаются. В таком случае возможна ситуация, когда пользователь не получит доступ в сеть, потому что он набрал пароль слишком хорошо. Чтобы этого не случилось, проводится следующая процедура. При каждом успешном входе в сеть осуществляется попытка автоматического занесения снятых биометрических данных в эталонную выборку, а самый первый элемент множества эталонов, характеризующих этого пользователя, удаляется из выборки.

Полученные от пользователей биометрические данные до применения процедуры распознавания предварительно фильтруются. Применение фильтров позволяет нивелировать разницу набора текста одним человеком и тем самым улучшить распознавание пользователей на коротких паролях (5-8 символов). В разработанном пакете программ используются как преобразование Фурье, так и эвристические алгоритмы.

Были проведены следующие эксперименты. Несколько человек с приблизительно равной скоростью набора русского текста ввели по 10-15 раз одну и ту же фразу из нескольких слов (до 30 символов) на русском языке без знаков пре-

пинания и пробелов. Общее число эталонов составило 83 элемента. Были получены следующие результаты, которые приведены в таблице 1, где используются следующие обозначения:

ЕЕ-4 – метод единственного эталона, работавший на l_p -метрике $l_p=4$.

ЕЕ-2 – метод единственного эталона, работавший на l_p -метрике $l_p=2$.

ДФФ – дискретное преобразование Фурье.

ЭАФ - эвристический алгоритм фильтрации.

Таблица 1

Длина пароля	Вероятность распознавания		
	Лучший метод	Лучший фильтр	Итого
31	ЕЕ-4, 91%	ДФФ, 91%	92%
30	ЕЕ-4, 95%	ДФФ, 91%	92%
25	ЕЕ-4, 91%	ДФФ, 91%	91%
20	ЕЕ-2, 93%	ДФФ, 91%	95%
15	ЕЕ-4, ЕЕ-2, 90%	ЭАФ, 86%	91%
10	ЕЕ-4, 90%	ЭАФ, 89%	92%
9	ЕЕ-4, 94%	ЭАФ, 95%	95%
8	ЕЕ-2, 92%	ЭАФ, 90%	91%
6	ЕЕ-4, 94%	ЭАФ, 89%	91%
5	ЕЕ-2, 91%	ЭАФ, 91%	91%

Подробные результаты в таблице 1 не приводятся из-за ограниченного объема данной статьи. В таблице 1 приведены только наилучшие результаты по трем категориям – лучший метод, лучший фильтр и общий результат.

Из анализа результатов следует, что для длинных паролей лучшим фильтром является ДФФ, а на коротких паролях лучше себя проявляют эвристические алгоритмы. Наблюдаемая нелинейность качества распознавания обусловлена тем, что наиболее точно индивидуальные особенности пользователя проявляются в переходах с одного слова фразы на другое, в то время как набор в пределах одного слова определяется опытом набора текста.

Таким образом, применение алгоритмов распознавания образов к системам аутентификации позволяет с высокой точностью аутентифицировать пользователей по их клавиатурному почерку. Качество распознавания одного пользователя достигает 95% даже для небольших паролей, поэтому разработанный алгоритм может быть применен как для контроля традиционной процедуры аутентификации, так и для замены парольных систем:

Литература

1. Фор А., Корман А., Денни-Папен М., Современная математика, М., Мир, 1986.
2. Фор А., Восприятие и распознавание образов, М., Машиностроение, 1989.

САМООБУЧАЮЩАЯСЯ НЕЙРОННАЯ СИСТЕМА ДЛЯ АВТОНОМНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Игнатюк О.Н.

Электронно-механический факультет,
Брестский политехнический институт, Московская 267,
224017 Брест, Республика Беларусь
cm@brpi.belpak.brest.by

Ключевые слова: нейронные сети, самообучение, мобильный робот

Введение

Самообучение характеризуется способностью системы обучаться при взаимодействии с внешней средой. В результате самообучения происходит самоорганизация системы с целью адаптации к внешней среде. Самообучение позволяет освободить оператора от процесса обучения и является важным фактором для эволюции системы. Так, при функционировании робота в разных условиях точность информации от сенсорных устройств может быть различной. В результате возникает необходимость корректировать знания, заложенные в систему, с целью адаптации к внешней среде. Особенно актуальным это является при функционировании робота в агрессивных средах или на других планетах, где невозможно предусмотреть все аспекты ситуационного взаимодействия робота с окружающей обстановкой. Рассмотрим основные принципы реализации концепции самообучения для мобильных роботов.

Общая архитектура системы

Общий подход к построению самообучающейся системы состоит в том, что начальные знания робота могут пополняться и корректироваться в процессе функционирования. При этом здесь предполагается, что базовые знания робота содержатся в блоках 1–3 и 6 (рис. 1), которые определяются логическим путем, как было показано в предыдущих разделах. Тогда задача состоит в том, чтобы в процессе функционирования робота обучить многослойный перцептрон (блок 4) для обеспечения робастного управления на узких интервалах движения. Схема

взаимодействия робота с внешней средой в процессе самообучения представлено на рис. 1.

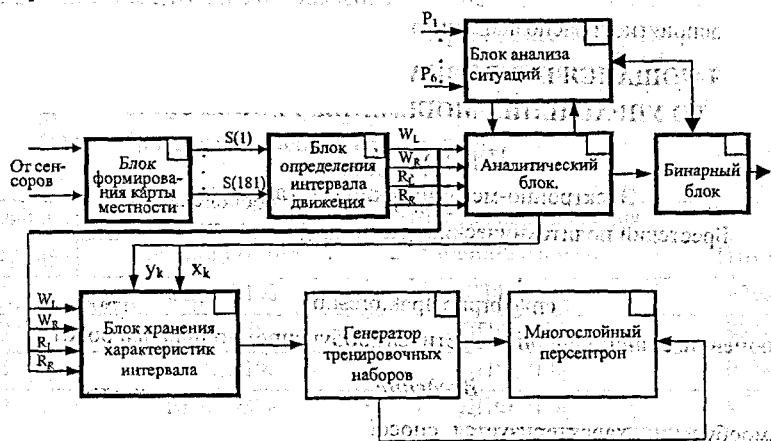


Рис 1. Архитектура нейронной системы в режиме самообучения

Управление при этом происходит от аналитического блока и бинарной нейронной сети. Процесс самообучения происходит методом проб и ошибок на узких интервалах движения. При успешном выполнении маневра формируются тренировочные наборы для обучения многослойного персептрона. При неудачной попытке происходит возвращение робота в исходную точку на несколько шагов назад и повторение маневра (рис. 2).

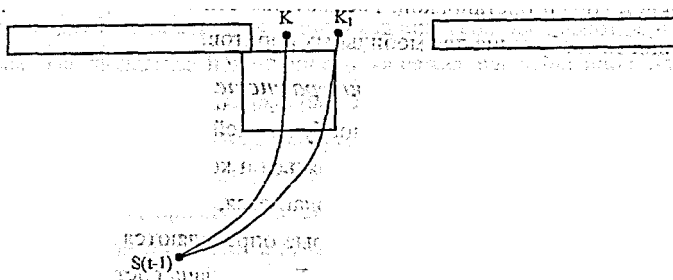


Рис. 2. Пример некорректной траектории робота (точка К): K_1 – скорректированное направление движения

Блок анализа ситуаций (рис. 1) предназначен для восстановления ситуации на предыдущем шаге робота $t-1$ и формирования корректирующего направления движения $\gamma(k_1)$:

$$\gamma(k_1) = \gamma(k) \pm \delta \quad (1)$$

где $\gamma(k)$ — направление движения, сформированное аналитическим блоком в данной точке на предыдущей попытке маневра; δ — угол коррекции направления движения.

В качестве входной информации блока анализа ситуации используются данные от тактильных датчиков (рис. 3).

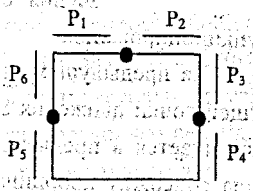


Рис.3 Расположение тактильных датчиков.

При этом выходной сигнал i -го датчика равняется единице $P_i = 1$, если произошел контакт соответствующего датчика с препятствием. В противном случае $P_i = 0$. Коррекция направления движения робота производится путем логического анализа информации от тактильных датчиков и предыдущего направления движения $\gamma(k)$:

$$(P_1 \vee P_6 = 1) \rightarrow \delta$$

$$(P_2 \vee P_3 = 1) \rightarrow -\delta$$

$$((P_4 = 1) \vee (P_5 = 1)) \wedge (\gamma = 1) \rightarrow -\delta$$

$$((P_4 = 1) \vee (P_5 = 1)) \wedge (\gamma = 0) \rightarrow \delta$$

В приведенных выше выражениях сигнал u формируется следующим образом:

$$u = \begin{cases} 1, & \text{если } \gamma(k) > 0 \\ 0, & \text{иначе} \end{cases} \quad (3)$$

Таким образом блок анализа ситуаций формирует положительное или отрицательное значение угла коррекции направления движения, если произошло столкновение с препятствием. Это означает, что точка K в выделенном интервале движения была выбрана неправильно и необходимо определить в соответствии с новым направлением $\gamma(k_1)$ координаты точки K_1 .

Так как бинарный блок работает под управлением аналитического блока, то в результате осуществляется также коррекция выходных данных бинарной нейронной сети. В некоторых ситуациях целесообразно корректировать выходные значения бинарной нейронной сети путем изменения варианта ее функционирования (рис. 1). Это осуществляется путем анализа информации от тактильных датчиков и предыдущих выходных значений бинарного блока. Например, если

$$(P_1 = 1) \wedge K(t-1) = 0110 \rightarrow K(t) = 0100,$$

$$(P_2 = 1) \wedge K(t-1) = 1010 \rightarrow K(t) = 1000,$$

где $K(t-1)$ и $K(t)$ – соответственно выходные значения бинарного блока на предыдущем и текущем этапе функционирования.

Для восстановления ситуации на предыдущем шаге блок анализа ситуации хранит координаты предыдущей точки движения $S(t-1)$. В случае столкновения с препятствием робот возвращается в предыдущую точку $S(t-1)$. Блок хранения характеристик интервала содержит координаты текущей точки K и расположение интервала движения (W_L, W_R, R_L, R_R). В случае успешного завершения маневра роботом они поступают в генератор тренировочных наборов. Маневр считается успешным, если робот достигает точки K в выделенном интервале движения без столкновений с препятствиями. В этом случае координаты точки K относительно подвижной системы координат робота равняются нулю.

Генератор на основе координат точки K и характеристик интервала движения путем вращения определяет для блоков MLP_1 и MLP_2 совокупность тренировочных наборов, количество которых равняется приблизительно 30. В результате моделирования различных ситуаций формируется обучающая выборка. Как показали эксперименты для устойчивой работы аналоговых нейронных сетей MLP_1 и MLP_2 необходимый объем обучающей выборки составляет 120 – 240 тренировочных наборов. В процессе функционирования робота в случае необходимости осуществляется также коррекция обучающей выборки для бинарного блока. Для устойчивой работы его достаточно 40 тренировочных наборов.

Применение бинарного блока для управления роботом в режиме самообучения позволяет уменьшить количество ошибок при совершении маневров и следовательно ускорить процесс самообучения. При этом самообучение может происходить как для получения новых, так и для коррекции старых знаний. В результате этого обеспечивается адаптация робота к внешней среде.

Экспериментальные результаты

Для тестирования самообучающейся системы было разработано программное обеспечение, позволяющее симулировать движение робота.

Были проведены тесты для различных ситуаций. Для обучения и симуляции были использованы некорректные значения, поступающие от датчиков:

- линейные расстояния до препятствий отличаются от реальных;
- угловые расстояния до препятствий отличаются от реальных;
- ошибки датчиков зависят от условий окружающей среды. Поэтому они изменяются с течением времени.

Для обучения многослойного перцептрона (блок 4) используется стандартный алгоритм обратного распространения ошибки с адаптивным шагом обучения [1]. Такой подход позволяет уменьшить время обучения. Эксперименты показали, что для генерации тестовых паттернов и обучения многослойного перцептрона требуется всего несколько секунд. Поэтому описываемая система функционирует в режиме реального времени.

После обучения движение робота было стабильным в различных ситуациях. Способность к самообучению позволяет роботу достигать цели в случае, когда ошибки датчиков изменяются в процессе движения. В результате этого робот адаптируется к окружающей среде.

Заключение

Этот доклад описывает определенный подход к вопросу управления мобильным роботом, согласно которому в процессе движения робота происходит самообучение и самоорганизация с использованием нейронных сетей. При взаимодействии робота с окружающей средой происходит сбор тренировочных данных, которые используются в последствии для обучения.

Литература

1. V. Golovko, J. Savitsky. Predicting neural net. In Proceedings Intern. Conf. CMNDT-95, Berlin, pp. 348-353.

СЛОВА СМЕЖНОГО КЛАССА КОДА ХЭММИНГА

Абрамук В.М., Маньяков Н.В., Махнист Л.П.

Брестский политехнический институт

224017, г. Брест, ул. Московская, 267

Предложен алгоритм генерации слов фиксированного веса смежного класса кода Хэмминга.

Ключевые слова: корректирующие коды, коды Хэмминга

В работе предложен алгоритм формирования двоичных последовательностей фиксированного веса кода Хэмминга [1]. Данный алгоритм предназначен для решения указанной задачи независимо от величины задаваемого веса, в отличие от алгоритмов, рассчитанных лишь для малых весов [2]. В его основу положен алгоритм генерирования множества $A(m)$ – всех кодовых слов весов от 0 до некоторого заданного четного числа k расширенного кода Хэмминга $H(m)$ длины $N = 2^m$:

begin

$A(1) := \{(00)\};$

$i := 2;$

$A(i) := \{\emptyset\};$

while $i < m$ *do*

if $A(i-1) = \{\emptyset\}$

then begin $i := i + 1; A(i) := \{\emptyset\};$ *end*

else

begin

$A(i-1) := A(i-1) \cup \{a\};$

if $w(\bar{a}) = 0$

then

begin

$p_0 := 0;$

repeat

“построить все вспомогательные вектора \bar{b} веса p_0 длины 2^{i-1} , которые содержат p_0 единиц в позициях, в которых вектор \bar{a} содержит нули, $A(i) := A(i) \cup \{\{\bar{b} | \bar{b} \text{ xor } \bar{a}\}\}$ ”;

$p_0 := p_0 + 2;$

until $p_0 > \min \{2(k \text{ div } 4), 2^{i-1}\};$

end!

else

for $p_0 := 0$ to $\min \{ (k-w(\bar{a})) \text{ div } 2, 2^{i-1}-w(\bar{a}) \}$ do

begin

if $p_0 \bmod 2 = 0$ then $p_1 := 0$ else $p_1 := 1$;

repeat

“построить все вспомогательные вектора \bar{b} веса p_0+p_1 длины 2^{i-1} , которые содержат p_0 единиц в позициях, где вектор \bar{a} содержит нули, и p_1 единиц в позициях, где вектор \bar{a} содержит единицы, $A(i) := A(i) \cup \{ \bar{b} | \bar{b} \text{ xor } \bar{a} \}$ ”;

$p_1 := p_1 + 2$;

until $p_1 > w(\bar{a})$;

end;

end;

end.

Количество векторов, полученных по вектору \bar{a} длины $n=2^{i-1}$ веса $w(\bar{a})$ на i -ом шаге алгоритма можно определить по формуле:

$$\sum_{j=0}^U C_n^{2j},$$

где $U = \min \{ (k \text{ div } 4), 2^{i-1} \text{ div } 2$, если $w(\bar{a}) = 0$; и

$$\sum_{j=0}^V \left(C_{n-w(a)}^{j'} \sum_l C_{w(a)}^{l'} \right),$$

где $V = \min \{ (k-w(\bar{a})) \text{ div } 2, 2^{i-1}-w(\bar{a}) \}$, а внутреннее суммирование производится по параметру l с шагом 2 от 0 до $w(\bar{a})$, если j – четно, и от 1 до $w(\bar{a})-1$, если j – нечетно, при $w(a) \neq 0$.

Количество кодовых слов весов от 0 до некоторого заданного четного числа k расширенного кода Хэмминга $H(m)$ длины $N = 2^m$ определяется соотношением [3]:

$$\sum_{j=0}^{\frac{k}{2}} C_N^{2j} + \frac{(-1)^j C_N^j (N_{\text{div } 2} - 1)}{2}$$

В основу приведенного алгоритма положено следующее свойство кодовых слов расширенного кода Хэмминга: кодовое слово длины 2^m дает возмож-

ность получить кодовые слова в два раза большей длины. Например, для слова $\bar{a}=0000$ веса $w(\bar{a})=0$ составляются все векторы \bar{b} четного веса этой длины: $\bar{b}=0000; 1100; 1010; 1001; 0110; 0101; 0011; 1111$. Тогда слова в два раза большей длины будут образованы по правилу:

$$(\bar{b} | \bar{b} \text{ xor } \bar{a}),$$

т.е. 00000000; 11001100; 10101010; и т. д.

Генерация всех слов расширенного кода Хэмминга $H(3)$ длины $N = 8$ приведена в таблице.

\bar{a}	$w(\bar{a})$	\bar{b}	$(\bar{b} \bar{b} \text{ xor } \bar{a})$
0000	0	0000	00000000
		1100	11001100
		1010	10101010
		1001	10011001
		0110	01100110
		0101	01010101
		0011	00110011
		1111	11111111
1111	4	0000	00001111
		1100	11000011
		1010	10100101
		1001	10010110
		0110	01101001
		0101	01011010
		0011	00111100
		1111	11110000

Для получения слов веса k на последнем шаге алгоритма, когда $i=p_0$ берут таким, чтобы выполнялось соотношение $w(\bar{a})+2p_0=k$. Слова заданного веса k кода Хэмминга получаются из векторов веса k соответствующего расширенного кода путем удаления нулевой координаты, если k — четно; и векторов веса $k+1$ — удалением координаты равной нулю, если k — нечетно.

Заметим, что идея данного алгоритма может быть использована при решении аналогичной задачи для примитивных кодов Боуза-Чоудхури-Хоквингема, исправляющих две ошибки [1].

Литература

1. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. Теория кодов, исправляющих ошибки: Пер. с англ. - М.: Связь, 1979. - 744 с.
2. Demidenko S., Piuri V., Ivanyukovich A. Error Localization in Test Outputs: a Generalized Analysis of Signature Compression // IEEE Second Asian Test Symposium (ATS-93). - Beijing, China, 1993. - P. 317-322.
3. Demidenko S., Ivanyukovich A., Makhnist L., Piuri V. On the Binary Sequences with Indistinguishable Signature for a Given Error Multiplicity in Electronic Testing // Journal of The Institution of Engineers. - Singapore, February, 1995, Vol. 35, N 1. - P. 63-66.

ВЫБОР АДАПТИВНОГО ШАГА ОБУЧЕНИЯ ПРИ ГРУППОВОМ ОБУЧЕНИИ ДЛЯ ЛИНЕЙНОЙ НЕЙРОННОЙ СЕТИ

Гладкий И. И.

Брестский политехнический институт

г. Брест ул. Московская 267

В статье рассмотрен случай выбора адаптивного шага обучения для линейной нейронной сети при групповом обучении. Получена формула вычисления адаптивного шага обучения.

Ключевые слова: нейронная сеть, адаптивный шаг обучения.

Рассмотрим линейную нейронную сеть, которая состоит из распределительного слоя нейронных элементов и выходного слоя. В качестве выходного слоя используются нейронные элементы с линейной функцией активации. Каждый нейрон распределительного слоя имеет синаптические связи со всеми нейронами обрабатывающего слоя. Выходное значение j -го нейрона сети определяется, как

$$y_j^k = \sum_i \omega_{ij} x_i^k - T_j \quad (1)$$

Среднеквадратичная ошибка сети для всей обучающей выборки будет равна

$$E = \frac{1}{2} \sum_k \sum_j (y_j^k - t_j^k)^2$$

Для нахождения адаптивного шага обучения будем использовать метод градиентного спуска.

$$a(t) = \min E \left\{ \left(\omega_{ij}(t) - a(t) \frac{\partial E}{\partial \omega_{ij}(t)} \right); \left(T_j(t) - a(t) \frac{\partial E}{\partial T_j(t)} \right) \right\}$$

Подставляя в (1) выражения для изменения весовых коэффициентов и порогов нейронных элементов

$$\begin{aligned} \omega_{ij}(t+1) &= \omega_{ij}(t) - a(t) \frac{\partial E}{\partial \omega_{ij}(t)}; \\ T_j(t+1) &= T_j(t) - a(t) \frac{\partial E}{\partial T_j(t)}. \end{aligned}$$

Получим

$$(y_j^k)' = \sum \omega_{ij}(t) x_i^k - T_j(t) - a(t) \left(\sum_i \frac{\partial E}{\partial \omega_{ij}(t)} x_i^k - \frac{\partial E}{\partial T_j(t)} \right) \quad (2)$$

Обозначим

$$a_j^k = \sum_i \frac{\partial E}{\partial \omega_{ij}(t)} x_i^k - \frac{\partial E}{\partial T_j(t)}$$

Тогда (2) примет вид

$$(y_j^k)' = y_j^k - a(t) a_j^k$$

Для определения адаптивного шага обучения необходимо найти такое значение $a(t)$, чтобы среднеквадратичная ошибка была минимальной

$$E = \frac{1}{2} \sum_{k,j} \sum ((y_j^k)' - t_j^k)^2 = \frac{1}{2} \sum_{k,j} \sum (y_j^k - a a_j^k - t_j^k)^2 \rightarrow \min$$

Тогда

$$\frac{\partial E}{\partial a} = \sum_k \sum_j (a(a_j^k)^2 - a_j^k (y_j^k - t_j^k)) = 0$$

Отсюда

$$a = \frac{\sum_k \sum_j a_j^k (y_j^k - t_j^k)}{\sum_k \sum_j (a_j^k)^2} \quad (3)$$

Так как $\frac{\partial^2 E}{\partial a^2} > 0$, то при данном a обеспечивается минимум среднеквадратичной ошибки.

Найдем выражение для a_j^k . Так как

$$\frac{\partial E}{\partial \omega_{ij}(t)} = \sum_p (y_j^p - t_j^p) x_i^p;$$

$$\frac{\partial E}{\partial T_j(t)} = -\sum_p (y_j^p - t_j^p);$$

$$a_j^k = \sum_p (y_j^p - t_j^p) (1 + \sum_i x_i^p x_i^k) \quad (4)$$

где $k = \overline{1, L}$; L - количество образов, подаваемых на вход сети при групповом обучении; x_i^k - i -ая компонента k -го образа.

Литература

1. В. А. Головки «Нейроинтеллект»: теория и применение. Книга 1: Организация и обучение нейронных сетей с прямыми и обратными связями. // Брест Изд. БПИ, 1999.

МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ ПРОДОЛЬНОГО ЗВУКА В СЛОИСТОЙ СТРУКТУРЕ, СОДЕРЖАЩЕЙ СЛОЙ ДИССИПАТИВНОЙ СРЕДЫ

Костюк Д. А.

Брестский политехнический институт
224017, Брест, ул. Московская, 267, Республика Беларусь

Теоретически рассмотрено нормальное распространение непрерывных и импульсных продольных акустических волн через зазор, заполненный сильно диссипативной средой (СДС), между двумя твердотельными полупространствами. Показана существенная зависимость коэффициентов прохождения и отражения и их фаз от коэффициента затухания продольного звука в среде зазора и от его безразмерной фазовой толщины. Программными средствами рассчитаны форма отраженного и прошедшего зазор акустического импульсного сигнала. Обсуждается использование полученных результатов к исследованию акустических свойств вязких жидкостей, а также диагностики технологических процессов.

Распространение объемных акустических волн в слоистых средах рассмотрено достаточно подробно, хотя аналитические решения найдены для трехслойной среды при нормальном к границам раздела сред падении волны и в предположении малого затухания звука в материалах, составляющих такую

слоистую структуру. Для наклонного падения волны на границы слоистой структуры, если того требуют практические приложения, необходимы компьютерные расчеты. В реальных экспериментах и технических применениях имеют дело не с непрерывными акустическими колебаниями, а с сигналами конечной, зачастую весьма короткой длительности и соответственно частотно широкополосными.

В твердых телах, да и в большинстве жидкостей за исключением резонансов взаимодействия упругих волн с другими элементарными возбуждениями вещества дисперсия скорости звука отсутствует вплоть до высоких частот. Тем не менее существуют сильно вязкие жидкости или смеси веществ при их химической реакции, в которых имеет место сильная дисперсия скорости звука.

Если в материалах, составляющих слоистую структуру, отсутствует дисперсия скорости звука, то отражение всех частотных составляющих импульсного сигнала от границы раздела сред происходит согласно классическим формулам Френеля, которые являются частотно независимыми, и следовательно спектр преобразованных сигналов не меняется. В рассматриваемом нами случае в силу существования дисперсии скорости звука в СДС разные частотные составляющие импульсного сигнала при его падении на границу раздела сред преобразуются различно и соответственно этому спектру отраженного от границы и прошедшего ее сигнала изменяются.

Пусть из твердого полупространства на СДС падает непрерывная гармоническая продольная волна (ПВ), которая частично отражается и проходит в жидкостный слой толщины d , а затем во второе твердое полупространство. Решения для ПВ в полупространствах ($x < 0$, $x > d$) и слое ($0 < x < d$) ищем в виде:

$$\begin{aligned} u_1 &= A_1 e^{i(k_1 x - \omega t)} + B_1 e^{i(-k_1 x - \omega t)}, \\ u &= A e^{i(kx - \omega t)} + B e^{i(-kx - \omega t)}, \\ u_2 &= A_2 e^{i(k_2 x - \omega t)} \end{aligned} \quad (1)$$

где $k_1 = \omega/S_{11}$, $k_2 = \omega/S_{21}$ — волновые числа, S_1 — скорость ПВ, ω — частота.

Решения (1) удовлетворяют соответствующим волновым уравнениям, а будучи подставленными в граничные условия, дают систему линейных уравнений для определения коэффициентов отражения $R = B_1/A_1$ и прохождения $T = A_2/A_1$:

$$R = \frac{z_0(z_1 - z_2)a + (z_1 z_2 - z_0^2 a^2)X}{z_0(z_1 + z_2)a + (z_1 z_2 + z_0^2 a^2)X} \quad (2)$$

$$T = \frac{4z_0z_2a}{z_0(z_1+z_2)aX_1 + (z_1z_2+z_0^2a^2)X_2} e^{-\alpha d} \quad (3)$$

где $z_0 = \rho_0 S_0$ — акустический импеданс слоя (при $\omega \rightarrow 0$), $a = 1 - ix + iy + x^2$, $x = \omega/\omega_0$, $\omega_0 = \rho_0 S_0^2 b$, b — параметр диссипативных потерь, $y = k''/k'$, $k' = k' + k''$ — волновое число, $X = X_1/X_2$, $X_1 = e^{-\beta} + e^{-\alpha} e^{i\beta}$, $\alpha = k''d$, $\beta = k'd$.

Коэффициент отражения при $\omega \rightarrow 0$, $R_0 = (z_1 - z_2)/(z_1 + z_2)$, а при $\omega \rightarrow R \rightarrow 1$. При изменении частоты, когда по толщине слоя укладывается $n\lambda/4$ длин волн, возникают минимумы и максимумы R_ω , т.е. возникают осцилляции коэффициента отражения ПВ. Коэффициент прохождения при $\omega \rightarrow 0$, $T_0 = 2z_1/(z_1 + z_2)$, а при $\omega \rightarrow T \rightarrow 1$.

Реальный импульсный акустический сигнал можно представить следующим выражением:

$$u'(x=0, t) = A e^{-\alpha t} \sin 2\pi \frac{t}{T} \left[\theta(t - \frac{\tau}{2}) - \theta(t + \frac{\tau}{2}) \right] \quad (4)$$

где α — безразмерный параметр, определяющий огибающую акустического сигнала; $T = 2\pi/\omega_0$, ω_0 — частота основной гармоники сигнала, $\tau = nT$ — длительность импульса, n — некоторое целое число, равное количеству периодов излучаемого импульса.

Исходя из приведенных зависимостей R_ω (2) и T_ω (3) и используя прямое и обратное преобразование Фурье с помощью компьютера рассчитывалась форма отраженного и прошедшего сигналов в вышеуказанных слоистых структурах.

(1) Заметим, что примененные программные средства позволяют выяснить особенности отражения, а также прохождения для любой формы излучаемых импульсов ПВ. Аналитические расчеты возможны частично для простейших форм излучаемых сигналов (например, для прямоугольного или нескольких периодов синусоидального, практически нереализуемых), но нетривиальный частотнозависимый вид R_ω и T_ω затрудняет или делает невозможным нахождение спектра и формы отраженных и прошедших сигналов.

Следует сделать вывод, что состояние отражающей СДС существенно влияет на коэффициент отражения и фазу как непрерывных, так и импульсных акустических сигналов. Так как фазовые измерения являются более точными по сравнению с амплитудными, то по ним мы можем судить о поглощении звука в диссипативной среде и проводить непосредственные измерения вязкости жидкостей. По таким измерениям при соответствующей доработке можно судить о готовности к употреблению того или иного технологического продукта. Так

можно контролировать качество сцепления асфальта с грунтом, бетона с железной арматурой, соответствие нормам при изготовлении композиционных и иных материалов, а также степень готовности пищевых продуктов при их приготовлении.

ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ РЕГИОНАЛЬНЫХ АГРОЭКОЛОГИЧЕСКИХ СИСТЕМ

Акулич Я.А., Лукша В.В., Шведовский П.В.

Брестский политехнический институт

Рассмотрены особенности исследования региональных агроэкологических систем, трансформируемых как из гео-, так и геозкосистем. При этом, системы рассмотрены с позиций цензурированных событий, развивающихся во времени.

Ключевые слова: моделирование, агроэкологические системы, события, развитие во времени, элементы, функционирование

Следует отметить, что моделирование региональных систем, как систем симметрично ветвящихся кратковременного централизованного действия с простыми переходами, не имеет больших перспектив. Даже использование моментных производящих функций позволяет осуществить только относительный анализ эффективности функционирования региональных систем.

Запишем моментную производящую функцию для группы элементов, подчиненных одному элементу $(N-1)$ ранга

$$\varphi(e^z) = (r_N \cdot e^z + q_N)^{n_N} \quad (1)$$

где Z_N – вероятность нормального состояния элемента N ранга; $q_N = 1 - r_N$.

Тогда моментная производящая функция для распределения числа нормально функционирующих элементов определится, в соответствии с формулой полной вероятности, в виде

$$\varphi_N(e^z) = \sum_{x=0}^{n_N-1} P_{n-1}(x) \cdot (r_N \cdot e^z + q_N)^{n_N}, \quad 0 \leq x \leq n_{N-1} \quad (2)$$

Используя правило дифференцирования сложной функции, имеем – $m_N^1 = m_{N-1}^1 \cdot n_N \cdot r_N$ или в замкнутой форме

$$m_N^1 = r_0 \cdot \Pi \cdot n_k \cdot r_k, \quad 1 \leq k \leq N \quad (3)$$

Так как случайное число нормально функционирующих элементов определяется как $V = \sum X_i, 1 \leq i \leq n_N$, то вероятность нормального функционирования системы, в целом, равна $\Phi(k) = 1 - (1-R)^k$, а полная вероятность

$$E = \sum P_N(k) \cdot \Phi(k); 1 \leq k \leq n_N \quad (4)$$

где k – число нормально функционирующих элементов; R – вероятность выполнения своей функции относительно нормального функционального состояния системы; $P_N(k)$ – вероятность того, что нормально функционирует ровно k элементов.

В развернутой форме имеем

$$E = 1 - \left\{ r_0 \cdot \left[r_1 \cdot \left(r_2 \dots r_{N-1} \cdot (r_N \cdot Q + q_N)^{n_N} + q_{N-1} \right)^{n_{N-1}} + \dots + q_1 \right] + q_0 \right\} \quad (5)$$

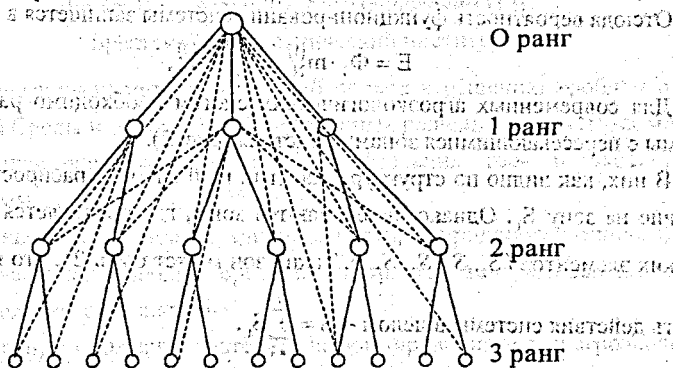


Рис. 1 Структура изотропной ветвящейся системы со сложными переходами.

Однако такая структура характерна только для однородных региональных или локальных геосистем (лесных, суходольных, простых ландшафтных и др.). Для неоднородных региональных наиболее характерна структура симметричных изотропных ветвящихся систем со сложными переходами (рис. 1).

Характерным для такой системы является возможность взаимного подчинения через несколько рангов, а также неполная идептичность элементов по параметрам надежности (выживаемости) и характеру ветвления. Для такой системы случайное число нормально функционирующих элементов N ранга при условии работоспособности (нормального функционирования) элементов 0 и 1 ранга (состояние "1,1") определяется по формуле

$$\xi_N = \xi_{N-1}^1 + \dots + \xi_{N-1}^k + \Theta_{N-1}^1 + \dots + \Theta_{N-1}^{n-k} \quad (6)$$

где Θ_N – случайное число нормально функционирующих элементов при условии, что нормально функционируют элементы 1 ранга и нарушено функционирование элементов 0 ранга, т.е. наблюдается состояние "1,0".

(4) Тогда моментная производящая функция распределения величины ξ_N определится следующим образом –

$$P_N(e^z) = [P_1 \cdot P_{N-1}(e^z) + q_1 \cdot R_{N-1}(e^z)]^n, \quad (7)$$

при $P_0(e^z) + Q_0(e^z) = e^z$; $R_0(e^z) = 1$, где $P_1 = (1 - \beta \cdot \gamma) \cdot (1 - \alpha)$; α – вероятность нарушения функционирования через взаимовлияние; β, γ – вероятность прямого и косвенного взаимовлияния, соответственно; $Q_0(e^z)$ и $R_0(e^z)$ – моментные производящие функции для состояний "0,1" и "1,0", соответственно.

Отсюда вероятность функционирования системы запишется в виде

$$E = \Phi_1 \cdot m_N^{(1)} + \Phi_2 \cdot m_N^{(2)}, \quad (8)$$

Для современных агроэкологических систем необходимо рассматривать системы с пересекающимися зонами действия (рис. 2).

В них, как видно по структуре, каждый i -ый элемент распространяет свое действие на зону S_i . Однако, наблюдаются зоны, где проявляется влияние нескольких элементов ($S_{23}, S_{12}, S_{34}, S_{234}$). Таких зон может быть 2^n , что и определяет

область действия системы в целом – $S = \sum_{i=1}^n S_i$.

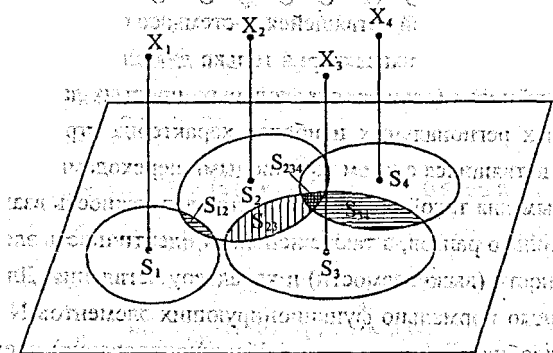


Рис. 2 Структура системы с пересекающимися зонами действия.

Тогда, соответственно, вероятность функционирования агроэкологической системы может быть определена по зависимости

$$E = \sum S_{\alpha\gamma} \cdot M \cdot \Phi_{\alpha\gamma}, \quad 1 \leq \gamma \leq 2n, \quad (9)$$

где $S_{\alpha\gamma}$ – работоспособность системы в зоне пересечения; $\Phi_{\alpha\gamma}$ – условная функция работоспособности для данной зоны при условии, что в рассматриваемый момент работоспособно все подмножество элементов α .

Литература

1. Шведовский П.В., Валуев В.Е. и др. Эколого-социальные аспекты освоения водно-земельных ресурсов и технологий управления режимами гидроме-лиорации. – Минск.: "Ураджай", 1998, 364с.

ОСОБЕННОСТИ ОПТИМИЗАЦИИ СТРУКТУРЫ И МОДЕЛИРОВАНИЯ ГЕНЕТИЧЕСКОЙ ЭВОЛЮЦИИ ГЕО-, ЭКО- И АГРОЭКОЛОГИЧЕСКИХ СИСТЕМ

Лукша В.В., Акулич Я.А., Шведовский П.В.

Брестский политехнический институт

Проанализирован методологический подход к решению проблем охраны Окружающей Среды и оптимизации инженерных решений и расчетных моделей в области природопользования и преобразования гео- и экосистем в агроэкосистеме.

Ключевые слова: модель, оптимальность, природопользование, формирование, управление, природная среда, структура, генетическая эволюция, гео-, эко- и агроэкологические системы

Сегодня, при множестве альтернатив формирования эко- и агроэкологических систем, крайне ограничены возможности анализа последствий этих преобразований как на ландшафты, так и Окружающую Среду, в целом.

Осуществление анализа на эвристическом уровне, когда учитываются только общие закономерности, без знания сложных транзитивных цепочек взаимодействий в системах, т.е. иерархической структуры и целостности, зачастую создает иллюзию решения проблем, которые в действительности могут перейти в надсистемы другого уровня (из локальных в региональные) или же переместиться в пространстве и сдвинуться во времени. Этим и определяется необходимость перехода с традиционного подхода – изучив основные процессы и взаимосвязи, объединить частные знания в общие, на нетрадиционный – из концептуального холистического (синтезирующего) рассмотрения проблемы уста-

новить интегрирующие и главенствующие приоритеты и возможности исследования частных составляющих компонент.

Нами предложена соответствующая понятийно-логическая модель гео- и агроэкологических систем вида $S = (Z, \Sigma, F, E, \Lambda)$; где S – упорядоченный набор множеств состояний и изменений; Z – множество целей преобразований (управлений); Σ – генетическая структура; F и E – функционирование в физическом и эволюция в геологическом времени, соответственно; Λ – эмерджентность, основу которой составляет вероятностно-стохастическая элементная структура, приведенная на рис. 1.



Рис. 1 Элементарная генетическая структура общих закономерностей эволюции систем.

Данная генетическая структура определяет и соответствующую базисную модель генетического формирования систем (рис. 2).

Разновидность ее элементов и структур, определяющие параметры которых имеют самый различный физический смысл и нестрогая определенность законов их влияния на функционирование системы в целом, обуславливают единственную возможность описания ее поведения – через случайную мерную функцию $x(t) = [x_1(t), \dots, x_n(t)]$.

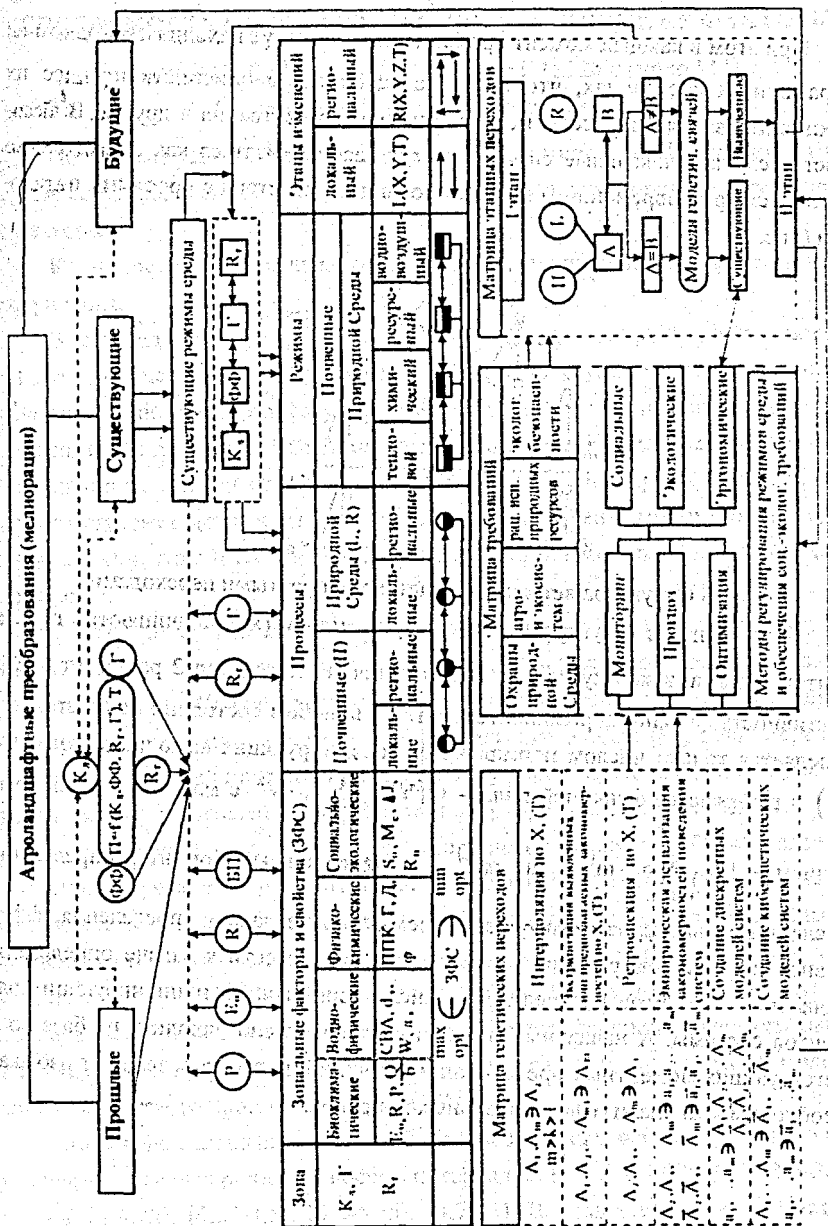


Рис. 2. Батисная модель генетического формирования агроландшафтных систем

При этом в каждый момент времени системы могут находиться в одном из 2^n различных состояниях, что позволяет собственно описывать процесс их функционирования процессом перехода из одного состояния в другое. В большинстве случаев локальные системы могут рассматриваться как симметрично ветвящиеся кратковременного централизованного действия с простыми переходами (рис. 3)

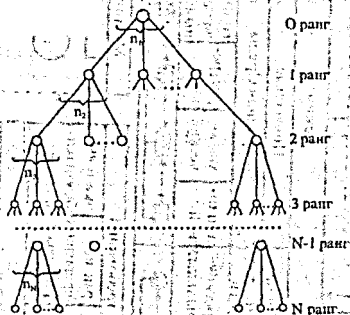


Рис. 3 Структура ветвящейся системы с простыми переходами.

Как видно из структуры, основному элементу (x_0) подчиняются n_1 элементов 1 ранга, каждому из них подчиняется n_2 элементов 2 ранга и т.д. Эффективность функционирования геосистемы в любом состоянии полностью определяется только числом нормально функционирующих выходных элементов (V) и выражается функцией вида $\Phi(V) = \sum_{1 \leq k \leq L} b_k \cdot V^k$ с математическим ожиданием $E = \sum_{1 \leq k \leq L} b_k \cdot m_N^{(k)}$, где $m_N^{(k)}$ — k -ый начальный момент распределения числа нормально функционирующих элементов, т.е. задачу определения эффективности функционирования геосистемы можно свести к задаче определения начальных моментов распределения числа нормально функционирующих элементов системы. А начальные моменты целесообразно находить на базе соответствующих моментных производящих функций, которые позволяют избежать необходимости анализа факториальных поправок.

УСПЕВАЕМОСТЬ СТУДЕНТОВ И ОПРЕДЕЛЯЮЩИЕ ЕЁ ФАКТОРЫ.

Яцкевич О.С., Соколюк Н.Л., Говорова Е.И., Дыдышко М.И.

Брестский политехнический институт

По данным успеваемости одного курса экономического факультета за три первых семестра, сделана попытка выяснить и проанализировать определяющие её факторы.

Ключевые слова: успеваемость, факторы, специальность, школа, уровень поступления.

В настоящей работе предлагаются первые результаты, связанные с попыткой дать количественный анализ успеваемости по группам различных специальностей экономического факультета Брестского политехнического института и выяснить факторы, оказывающие воздействие на неё. Были рассмотрены экзаменационные оценки студентов второго курса за три первых семестра. Для каждого студента вычислены средние оценки за каждый семестр и общая средняя за все семестры и на основании этих величин проводилось дальнейшее исследование.

В качестве признаков-факторов решено было выделить следующие: пол студента, школа, место проживания, специальность и уровень поступления. Поясним подробнее. **Школа:** среди школ, которые окончили студенты, были выделены три уровня: лицеи и гимназии (СП), сельские школы (С), общие средние школы (СШ). **Место проживания** (во время учёбы): здесь также выделено три уровня — дома у родителей (Д), на частной квартире (К), в общежитии (О). **Уровень поступления:** пять уровней — прошедшие по конкурсу (К); зачисленные на платное обучение по результатам вступительных экзаменов (НК); не выдержавшие все экзамены, но зачисленные на платное обучение по результатам собеседования (С); окончившие подготовительное отделение института (О); переведённые из других ВУЗов (П). **Специальность:** анализ проведён по четырём специальностям экономического факультета, то есть по специальностям маркетинг (МА), бухгалтерский учёт (БУ), коммерческая деятельность (КД), мировая экономика и международные отношения (МО).

При анализе данных отрезок от 3 до 5 возможных значений среднего балла успеваемости был разбит на восемь интервалов: [3-3,25), [3,25-3,5), [3,5-3,75), [3,75-4), [4-4,25), [4,25-4,5), [4,5-4,75), [4,75-5]. В дальнейшем первые четыре интервала будем называть условно плохими, а остальные четыре — хорошими.

Представим некоторые результаты анализа. Прежде всего, отметим, что все выше названные признаки действительно в среднем являются определяющими для успеваемости студентов и позволяют сделать некоторые организационные выводы. Рассмотрим, как влияет на успеваемость каждый из этих факторов.

Фактор «Пол». По всем семестрам на хороших интервалах больший процент у девушек, что отражается в общих результатах. Он составляет 69,272%, соответственно у юношей – 42,029%. Отсюда следует, что безусловными лидерами на плохих интервалах являются юноши. Их процент составил 57,970%, когда у девушек эта цифра всего лишь 30,728%.

Фактор «Учебные заведения» отражается на успеваемости студентов следующим образом: студенты, ранее обучавшиеся в спецшколах, в основном оправдывают высокий уровень подготовки, о чём свидетельствует их большой процент на хороших интервалах. Он составил 82,937%. Соответственно, на плохих интервалах эта цифра будет очень мала, всего 17,063%. Студенты, окончившие обычные средние школы, показывают более низкий уровень знаний. Всего 63,214% студентов сдали все три сессии без троек. Студенты из сельских школ показывают ещё более низкий уровень подготовки. 51,515% этих студентов равномерно распределены на плохих интервалах, то есть без троек сдали все три сессии 48,485% студентов.

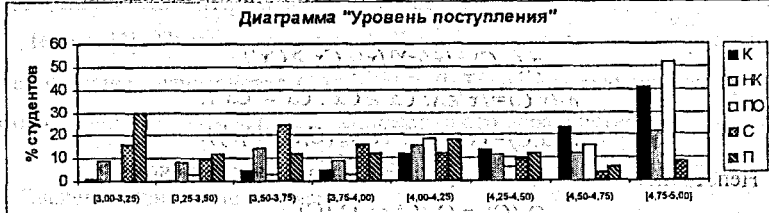
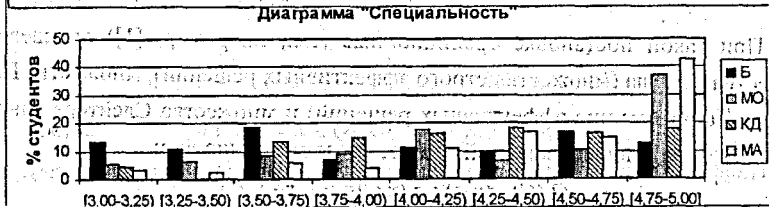
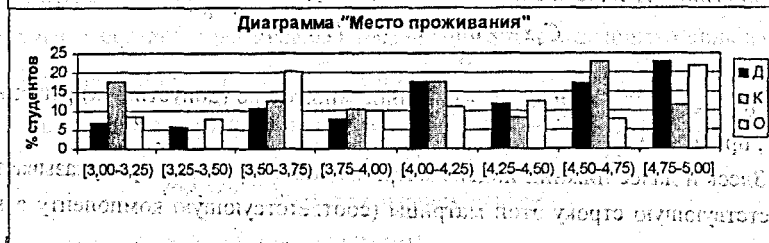
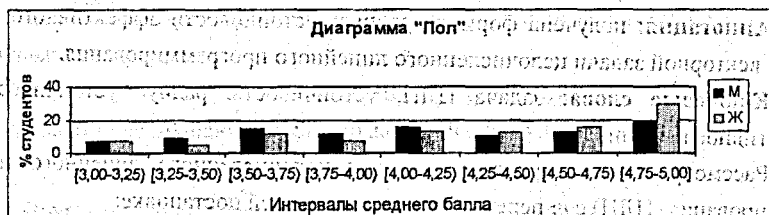
Фактор «Место проживания» имеет следующее влияние на успеваемость: студенты, которые живут дома, показывают лучшую успеваемость, чем студенты живущие в общежитиях и на квартирах, что хорошо отражено в следующих данных. На хороших интервалах преобладают студенты, проживающие дома – 69,168%, процент студентов, живущих на квартирах меньше – 59,848; а процент студентов из общежитий составляет всего 53,139%.

Фактор «Уровень поступления» влияет на успеваемость следующим образом: наилучшая успеваемость у тех, кто прошёл по конкурсу – 89,418% из них имеют средний балл выше 4, и у поступивших с подготовительного отделения, средний балл выше 4 среди них имеют 90,909% студентов. Процент студентов, которые не прошли по конкурсу, а были зачислены по результатам экзаменов, распределён по всем интервалам более или менее равномерно: 40,487% на хороших интервалах, 59,513% на плохих. Те, кто поступил по собеседованию и студенты переведённые из других ВУЗов показывают худшую успеваемость. Их проценты на плохих интервалах составляют 66,049% и 64,706% соответственно.

Фактор «Специальность». Наилучшая успеваемость у студентов из групп маркетинга, наихудшая – из групп бухгалтерского учёта. Так на хорошем интервале, проценты студентов всех специальностей равны соответственно: МА – 84,35% МО – 70,765% КД – 67,739% БУ – 50,193%

Самый маленький процент студентов из групп БУ можно объяснить большим числом студентов, поступивших по собеседованию и не прошедших по конкурсу.

Следует отметить ещё один настораживающий факт: средняя успеваемость студентов падает от семестра к семестру, что хорошо прослеживается из динамики средних баллов юношей и девушек: для юношей $x_1=4,1422$; $x_2=4,1062$; $x_3=4,0654$; и для девушек $y_1=4,241$; $y_2=4,1954$; $y_3=4,183$.



В завершение приведём итоговые диаграммы распределения среднего балла в зависимости от признак-фактора.

Авторы надеются, что дальнейшее более детальное изучение этих вопросов даст ощутимые результаты для практического воплощения их.

О РАДИУСЕ УСТОЙЧИВОСТИ ЭФФЕКТИВНОГО РЕШЕНИЯ ВЕКТОРНОЙ ЗАДАЧИ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Никулин Ю.В.

Белорусский государственный университет

220050 Минск, пр. Ф. Скорины, 4, e-mail: eva@mmf.bsu.unibel.by

Аннотация: получена формула радиуса устойчивости эффективного решения векторной задачи целочисленного линейного программирования.

Ключевые слова: задача ЦЛП, устойчивость, радиус устойчивости, эффективное решение.

Рассмотрим n -критериальную задачу целочисленного линейного программирования (ЦЛП) с m переменными в следующей постановке:

$$C_i x^* \rightarrow \min_{X}, \quad i \in N_n := \{1, 2, \dots, n\},$$

где $C = [c_{ij}]_{n \times m} \in R^{n \times m}$, $m, n \in N$, X – конечное множество (допустимых) решений в Z^m , причем $|X| > 1$.

Здесь и далее нижний индекс матрицы (так же как вектора) указывает на соответствующую строку этой матрицы (соответствующую компоненту вектора).

При такой постановке традиционные (см., например, [1]) определения множества Смейла (множества строго эффективных решений), множества Парето (множества истинно эффективных решений) и множества Слейтера (множества слабо эффективных решений) имеют соответственно вид:

$$Q_1(C) = \{x \in X: q_r(x, C) = \emptyset, \quad r = 1, 2, 3\},$$

где

$$q_1(x, C) = \{x' \in X \setminus \{x\}: Cx \geq Cx'\},$$

$$q_2(x, C) = \{x' \in X: Cx \geq Cx', \quad Cx \neq Cx'\},$$

$$q_3(x, C) = \{x' \in X: \forall i \in N_n (Cx > Cx')\}.$$

Непосредственно из этих определений вытекают включения

$$Q_1(C) \subseteq Q_2(C) \subseteq Q_3(C) \quad (1)$$

Заметим, что множество $Q_1(C)$ может быть пустым.

Для всякого числа $r \in N_3$ задачу поиска множества эффективных решений $Q_r(C)$ будем называть векторной (n -критериальной) задачей ЦЛП и обозначать через $Z_r^n(C)$, $n \geq 1$.

Для всякого числа $k \in N$ в k -мерном действительном пространстве R^k зададим чебышевскую норму $\|z\| = \max\{|z_i| : i \in N_k\}$, а в пространстве, сопряженном с R^k , — норму $\|z\| = \sum_{i \in N_k} |z_i|$.

При этом под нормой матрицы $B = [b_{ij}]_{n \times m} \in R^{nm}$, будем понимать норму вектора $(b_{11}, b_{12}, \dots, b_{nm})$.

Для всякого фиксированного числа $r \in N_3$ решение $x \in Q_r(C)$ назовем устойчивым, если

$$\exists \varepsilon > 0, \forall B \in \mathcal{H}(\varepsilon) (x \in Q_r(C+B)),$$

где множество возмущающих матриц задается формулой

$$\mathcal{H}(\varepsilon) = \{B \in R^{nm} : \|B\| < \varepsilon\}.$$

Другими словами, решение $x \in Q_r(C)$ устойчиво, если найдется такое число $\varepsilon > 0$, что x остается эффективным решением соответствующего вида в любой возмущенной задаче $Z_r^n(C+B)$, $B \in \mathcal{H}(\varepsilon)$.

Радиусом устойчивости решения $x \in Q_r(C)$ назовем число

$$\rho_r^n(x, C) = \begin{cases} \sup \Omega_r(x, C), & \text{если } \Omega_r(x, C) \neq \emptyset, \\ 0, & \text{в противном случае,} \end{cases}$$

где $\Omega_r(x, C) = \{\varepsilon > 0 : \forall B \in \mathcal{H}(\varepsilon) (x \in Q_r(C+B))\}$.

В силу включения (1) очевидны неравенства

$$\forall n \in N \quad \forall x \in Q_1(C) \quad (\rho_1^n(x, C) \leq \rho_2^n(x, C) \leq \rho_3^n(x, C)),$$

$$\forall n \in N \quad \forall x \in Q_2(C) \quad (\rho_2^n(x, C) \leq \rho_3^n(x, C)).$$

Для того, чтобы вывести формулу радиуса устойчивости эффективного решения, нам понадобится следующая известная

Лемма [2]. Пусть векторы $x, x' \in X$ таковы, что для некоторого индекса $i \in N_n$ справедливо неравенство $C_i(x' - x) > 0$. Тогда для всякого вектора $b \in R^m$ такого, что $\|b\| \cdot \|x' - x\|^* < C_i(x' - x)$, имеет место неравенство

$$(C_i + b)(x' - x) > 0.$$

Введем обозначение

$$\varphi^n(x, C) := \min_{x' \in X(x)} \max_{i \in N_n} \frac{C_i(x' - x)}{\|x' - x\|^*}$$

Очевидны утверждения:

$$\forall x \in Q_3(C) \quad \forall C \in R^{nm} \quad (\varphi^n(x, C) \geq 0, x \in Q_1(C) \Leftrightarrow \varphi^n(x, C) > 0) \quad (2)$$

Теорема. При любых числах $r \in N_3$, $n \in N$ и любой матрице $C \in R^{nm}$ для радиуса устойчивости всякого решения $x \in Q_1(C)$ векторной задачи ЦЛП $Z_r^n(C)$ справедлива формула

$$\rho_r^n(x, C) = \varphi^n(x, C).$$

Доказательство. Пусть $C \in R^{nm}$, r — фиксированное число из множества N_3 , $x \in Q_1(C)$, $\varphi := \varphi^n(x, C)$.

Сначала докажем неравенство $\rho_r^n(x, C) \geq \varphi$.

Если $\varphi = 0$, то это неравенство очевидно. Пусть $\varphi > 0$. Тогда в силу (2) $x \in Q_1(C)$. Пусть возмущающая матрица $B \in \mathcal{M}(\varphi)$. Согласно определению числа φ для любого решения $x' \in X(x)$ существует такой индекс $i \in N_n$, что выполняются неравенства

$$\frac{C_i(x' - x)}{\|x' - x\|^*} \geq \varphi > \|B_i\|.$$

Отсюда по лемме имеем

$$(C_i + B_i)(x' - x) > 0.$$

Поэтому с учетом определения множества Смейла находим $x \in Q_1(C+B)$. Следовательно (ввиду (1)), заключаем

$$\forall r \in N_3 \quad \forall B \in \mathcal{M}(\varphi) \quad (x \in Q_1(C+B)).$$

Откуда и выводим неравенство $\rho_r^n(x, C) \geq \varphi$. Далее докажем неравенство $\rho_r^n(x, C) \leq \varphi$. Пусть $\varepsilon > \varphi$. Тогда согласно определению числа φ справедлива формула

$$\exists x' \in X(x) \quad \forall i \in N_n \quad (\varphi \geq \frac{C_i(x' - x)}{\|x' - x\|^*}) \quad (3)$$

Рассмотрим возмущающую матрицу $B = [b_{ij}]_{n \times m}$, элементы которой для любого индекса $i \in N_n$ задаются по формуле

$$b_{ij} = \begin{cases} \alpha & \text{при } x'_j \leq x_j, \\ -\alpha & \text{при } x'_j > x_j, \end{cases}$$

где $\varphi < \alpha < \varepsilon$. Очевидно, что $B \in \mathcal{M}(\varepsilon)$.

Далее с учетом неравенств (3) и строения матрицы B легко убедиться в справедливости для любого индекса $i \in N_n$ следующих соотношений

$$(C_i + B_i)(x' - x) = C_i(x' - x) + B_i(x' - x) = C_i(x' - x) - \alpha \|x' - x\| \leq < C_i(x' - x) - \varphi \|x' - x\| < 0.$$

Поэтому $x \notin Q_i(C+B)$.

Итак, для любого числа $\varepsilon > \varphi$ существует такая матрица $B \in \mathcal{M}(\varepsilon)$, что $x \in Q_i(C+B)$. Поэтому для всякого числа $\varepsilon > \varphi$ выполняется неравенство $\rho_r^n(x, C) < \varepsilon$.

Следовательно, справедливо неравенство $\rho_r^n(x, C) \leq \varphi$.

Собирая все сказанное, убеждаемся в справедливости теоремы.

Литература

1. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задпч. М.: Наука. 1982.
2. Емеличев В.А., Подкопаев Д.П. // Журнал вычислительной математики и математической физики. 1998. Т.38. № 11. С.1801-1805.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ НАПРЯЖЕННОГО СОСТОЯНИЯ В МАССИВЕ В ОКРЕСТНОСТИ ВЫРАБОТОК БОЛЬШИХ ПРОТЯЖЕННОСТЕЙ.

Лазуренко Ю. Н.

Белорусский государственный университет
Пр. Ф. Скорины 4, 220050 г. Минск, Беларусь.

В докладе описывается некоторый подход к моделированию геомеханических процессов в массиве, нарушенном проведением выработок больших протяженностей.

Ключевые слова: выработка, лава, «длинный очистной забой», целик.

При выполнении модельных исследований по изучению геомеханического состояния массива в окрестности выработанного пространства, представляющего собой результат использования технологических схем обработки «длинными очистными забоями», одной из главнейших является проблема корректной формулировки модельных задач.

Формулировка модельной задачи включает в себя обоснование геомеханической схемы развития деформационных процессов в окрестности вырабо-

танного пространства, построение граничной модельной задачи, выбор метода решения и др.

Для наиболее полного анализа геомеханического поведения массива в окрестности отработанного пространства необходимо построение нескольких расчетных моделей, описывающих различные возможные схемы развития деформационных процессов и различные этапы этих процессов (во временной и пространственной протяженностях).

Выбор модели должен производиться таким образом, чтобы количественный или качественный анализ выбранных характеристик и свойств с помощью современных средств исследования можно было провести экономно по затратам времени и труда и с различной точностью.

При моделировании геомеханических процессов необходимо руководствоваться следующими положениями [1]:

В первую очередь необходимо построить модель геомеханического состояния изучаемой области массива, как механической среды.

Будем рассматривать массив как слоистое вязкоупругое тело, с учетом появления элементов разрушения при нарушении выполнения критериев прочности в массиве.

Так как моделируемая область породного массива является существенно неоднородной, то необходимые для расчетов физико-механические характеристики являются эффективными величинами, полученными расчетным образом [1,4,5].

- Удельный вес породной толщи - $20,48 \text{ кН/м}^3$;
- Модуль деформации непосредственной кровли - $12 \cdot 10^6 \text{ кПа}$;
- Модуль деформации породной толщи - 10^6 кПа ;
- Коэффициент Пуассона породной толщи - $0,35$;
- Предел прочности на сжатие пород выемочного пространства - 280 кПа ;
- Предел прочности на сжатие породной толщи - 270 кПа ;
- Удельное сцепление породной толщи - 1300 кПа ;
- Угол внутреннего трения - 79° ;
- Граничный угол свода давления (обрушения) - 55° .

Следующим этапом является построение в рамках выбранной модели массива расчетных схем, описывающих изменение напряженно-деформационного состояния областей массива в окрестности подземных сооружений.

Изучение процессов развития напряженно-деформированного состояния в породной толще вследствие наличия в последнем подземных выработок изучаемого технологического пространства выполнялось с целью исследования распространения возмущений в целиках и высоты развития магистральных трещин в кровле выработок (рис 1.1).

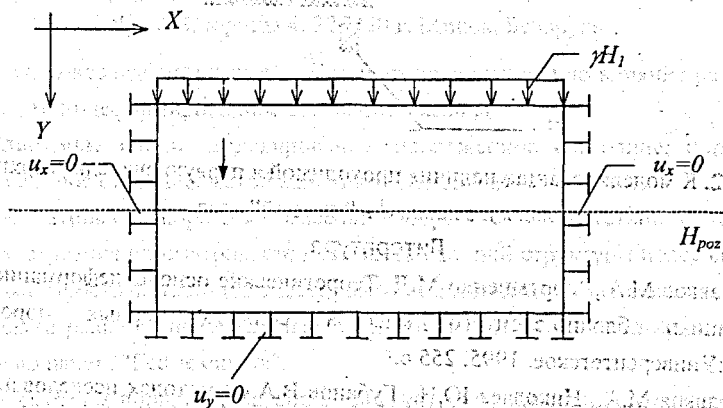


Рис.1.1. Базовая расчетная схема породного массива

Далее опишем математическую модель состояния породной толще при использовании различных предложенных технологических схем отработки системами смежных лав. При проведении численного моделирования учитываем последовательность по времени проведения выработок в массиве.

Особенностью выполнения модельных расчетов является наличие "верхних" (пройденных в первую очередь) и "нижних" лав. Технология отработки "верхних" лав предусматривает полное обрушение пород в кровле лавы [2,3]. Поэтому, наличие "верхней" лавы с учетом факторов ее первичной (по отношению к "нижней") отработки и полного обрушения кровли моделировалось наличием в рассматриваемой области элементов с разрушенной породой, то есть с отличными от физико-механических свойств основного массива свойствами (модулем упругости, коэффициентами сцепления и внутреннего трения). Наличие закладочного материала в "нижней" лаве моделировалось включениями элементами с характерными для поведения закладочного материала механическими свойствами (рис 1.2).

В результате численных экспериментов были выбраны наиболее предпочтительные из числа предложенных технологические схемы.

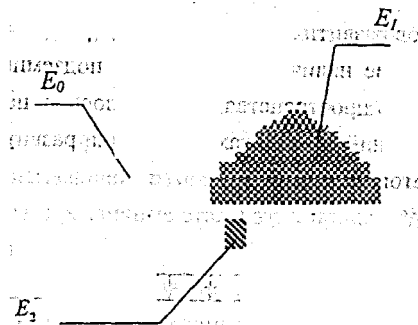


Рис. 1.2. К моделированию наличия проходимой в первую очередь "верхней" лавы и закладки в "нижней" лаве

Литература.

1. Журавков М.А., Мартыненко М.Д. Теоретические основы деформационной механики блочно-слоистого массива соляных горных пород. - Мн.: Университетское, 1995. 255 с.
2. Журавков М.А., Николаев Ю.Н., Губанов В.А. О методах исследования закономерностей формирования и распределения опорного давления при разработке калийных месторождений, пологого залегания. - Мн.: Изд-во БелНИИНТИ, 1991. 24 с.
3. Галаев И.З., Иванов А.А. Управление горным давлением при разработке рудных месторождений системами с открытым очистным пространством. - Л.: Изд-во ЛПИ, 1986. 84 с.
4. Барях А.А., Константинова С.А., Асанов В.А. Деформирование соляных пород. - Екатеринбург: Изд-во УрО РАН, 1996. - 203 с.
5. Испытание и внедрение способа охраны выработок с помощью компенсационных щелей и разработка научных основ прогноза деформаций выработок главных направлений на ПО "Беларуськалий". Методика прогноза состояния капитальных выработок на калийных рудниках ПО "Беларуськалий". // Отчет БФ ВНИИГ 12.С.88-170.88, х/д, №ГР01880022498. - Солигорск. 1988.

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ НАПРЯЖЕННОГО СОСТОЯНИЯ В ОБЛАСТЯХ ПОРОДНОГО МАССИВА С ГЕОЛОГИЧЕСКИМИ НАРУШЕНИЯМИ

Стагурова О. В.

Белорусский государственный университет
Пр. Ф. Скорины 4, 220050 г. Минск, Беларусь

Научно-исследовательская работа посвящена изучению влияния разломов на напряженно-деформированное состояние массива.

Ключевые слова: моделирование, напряженное состояние, породный массив, блочное строение, метод конечных элементов.

Рассматриваем породный массив, который в соответствии с теорией блочного строения рассматривается как часть блочной структуры более высоко-го порядка. Пространственную задачу разбиваем на несколько плоских.

Задачи решаем универсальным численным методом конечных элементов с помощью пакета "Геомеханика".

Рассмотрим породный массив в плоскости, параллельной земной поверхности. Моделируем численные расчетные схемы, соответствующие естественному состоянию массива с различными видами геологических нарушений (разрезов) и не противоречащие экспериментальным результатам (рис. 1).

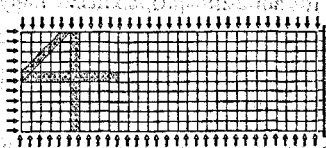


Рис. 1

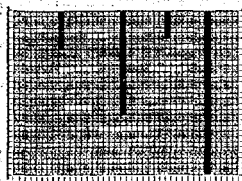


Рис. 2

По решенным задачам делаем выводы. Во всех случаях можно сделать заключение: при введении в массив разреза, поле главных напряжений изменяется, становится неоднородно, главные напряжения поворачиваются на некоторый угол. Наблюдается симметричность главных напряжений относительно серединной нормали к разрезу. Внутри разреза главные напряжения ориентируются таким образом, чтобы одна из компонент напряжений была ортогональна нормали разреза. Изменение ориентации напряжений не всегда ведет к изменению величин напряжений и наоборот, изменение значений напряжений не всегда влечет за собой поворот главных напряжений. Влияние разрезов распространя-

ется на расстояние в несколько раз превосходящее длину разреза. Максимальные величины напряжений наблюдаются при динамических передвижках краев разрезов. При этом в массиве наблюдаются не только деформации сжатия, но и растяжения. Увеличение ширины разреза влечет за собой меньшую дифференциацию значений напряжений. Условие полного сцепления приводит к формированию только деформации сжатия, и, как следствие, к повышению напряжений на границах разреза. Наибольшие напряжения возникают на краях разреза, в торцах и на пересечениях разрезов. Характерной чертой является повышение σ_1 с одновременным понижением σ_3 .

В работе рассматривалось НДС массива с разрывными нарушениями в поле силы тяжести.

В соответствии с исходной геомеханической моделью была смоделирована численная расчетная схема массива, находящегося под действием сил тяжести с объемным весом γ , с учетом влияния техногенного воздействия (рис.2).

В результате выполнения численных исследований получаем характеристики полного тензора напряжений и наведенного поля перемещений. Естественно, в зонах геологических нарушений (например, разрывных нарушений) литостатическое напряженное состояние нарушается.

Зоны разрывных нарушений моделируются сжимаемой упругой средой. Контактные условия между массивом и зонами разрывных нарушений моделируются условиями от полного сцепления до упругого взаимодействия.

При анализе результатов математического моделирования основное внимание уделялось: изучению картин изменения напряженного состояния породного массива в выделенной области при различных условиях и соотношениях динамических движений разрывных зон и выработанного пространства и возможности возникновения динамических срывов (появления динамических подвижек) в зоне разрывного нарушения.

Проведенные расчеты показали, что значение максимального вертикального главного напряжения σ_1 изменяется незначительно, то есть создание подземных крупномасштабных сооружений практически не оказывает влияния на величину σ_1 . Изменение затрагивает величины компонент главных субгоризонтальных напряжений σ_2 и σ_3 .

Важным является то, что перераспределение величин субгоризонтальных главных напряжений влечет за собой изменение и значений касательных напря-

жений $\sigma_{\text{max}} = (\sigma_3 - \sigma_1)/2$. Последнее обстоятельство является существенным для активизации движений породных массивов в горизонтальных плоскостях.

Расчеты показали, что в случае отсутствия в рассматриваемой области породного массива зон разрывных нарушений, изменения начального напряженного состояния в массиве главным образом определяются параметром "НЛ" - отношением глубины ведения подземных горных работ к характерному линейному размеру зоны освоения подземного пространства (выбирается наибольший из всех существующих).

Изменения напряженного состояния могут быть существенными в случае ведения крупномасштабных подземных работ в приповерхностных областях (на глубинах до 100 метров).

В случае же освоения подземного пространства в заглубленных областях и на глубоких горизонтах изменения величин тензора напряжений не являются такими, чтобы повлечь за собой возникновение значительных разрушений породных массивов или возникновение динамических срывов.

Иная картина наблюдается в области наличия зон разрывных нарушений. Воздействие крупномасштабной подземной разработки массива на зоны с разрывными нарушениями обуславливает потенциальную возможность возникновения техногенных динамических срывов (сейсмических явлений).

С точки зрения основных законов механики наиболее опасными (как потенциальная возможность возникновения динамических разрушений) являются большие сдвиговые относительные (по отношению друг к другу) вертикальные Δu_i и горизонтальные Δu_j , Δu_k подвижки граничных поверхностей блоков массива.

В первом приближении были сделаны расчеты для Солигорского месторождения.

В работе, наряду с изучением влияния геологических нарушений на НДС породного массива, было показано, что можно моделировать абстрактные задачи геомеханики и успешно их решать численными методами, а на базе полученных результатов исследований решать конкретные практические задачи.

Литература

1. Адушкин В.В., Цветков В.М. Влияние структуры и геодинамики на напряженное состояние земной коры. // XI Российская конференция по механике горных пород. Санкт-Петербург. 1977. С.7-12.

2. Усаченко Б.М., Перепелица В.Г.; Оценка напряженного состояния породного массива вблизи геологических нарушений. // Известия ВУЗов. Горный журнал. №8. 1990. С.37 - 40.
3. Дашко Р.Э. Механика горных пород. М.: Недра. 1987. 263 с.

МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ ДЕНЕЖНОГО РЫНКА НА ОСНОВЕ УРАВНЕНИЯ МАРШАЛЛА-ПИГУ

Народецкий Д. А.

Белорусский государственный университет
г. Минск, пр. Скорины 4

Аннотация: вопрос "достаточно ли денег в экономике" является одной из центральных проблем денежно-кредитного регулирования.

Для моделирования функционирования денежного рынка используется неоклассический подход. В работе построено семейство функций спроса на деньги; проведен их сравнительный анализ; из возможных факторов, определяющих спрос на деньги, выделен основной; рассмотрены двухфакторные функции спроса на деньги.

Ключевые слова: спрос на деньги; денежный рынок; линейная регрессия.

Под рынком денег в макроэкономической теории понимается совокупность отношений между банковской системой, создающей всеобщие платежные средства, и другими макроэкономическими субъектами, предъявляющими спрос на них. Изучение спроса на деньги является важной задачей с точки зрения оценивания возможного объема предложения денег, поскольку, как хорошо известно, необоснованное увеличение денежной массы приводит к инфляции.

Для моделирования функционирования денежного рынка используется неоклассический подход, в соответствии с которым спрос на деньги определяется потребностями субъектов в деньгах для осуществления покупок и платежей в промежутках между моментами получения денежных доходов.

За основу при построении модели было принято кембриджское уравнение Маршалла-Пигу:

$$M = kPy, \quad (1)$$

где M - некоторый денежный агрегат, k - коэффициент «предпочтения ликвидности»; P - индекс цен, y - количество транзакций в экономике, измеряемое обычно объемом реального дохода или производства. Разделив на индекс

цен и прологарифмировав, представим (1) в виде модели линейной регрессии:

$$\ln m_t = a_0 + a_1 t + a_2 \ln y_t + \varepsilon_t \quad (2)$$

Линейный тренд введен в качестве заменителя каких-либо неучтенных факторов, а также как показатель тенденций. В работе были рассмотрены три показателя – претендента на роль переменной y в уравнении (2):

- ВВП (GDP);
- средняя зарплата в народном хозяйстве (w);
- объем промышленного производства (Q);

Для каждого показателя была оценена регрессия. Данные для статистического анализа были взяты из ежемесячника «Статистический бюллетень» за 1998 год. Анализ проводился при помощи пакета «Стан-У». Наиболее адекватной моделью оказалась та, в которой в качестве основного фактора, определяющего объем спроса на деньги, была ставка средней заработной платы. Ниже приведены полученные уравнения для агрегатов $M0$ и $M2$:

$$\begin{aligned} a_0 = 0.755 & \quad a_1 = 0.122 & \quad a_2 = 0.00246 & \quad R^2 = 0.798 \\ [0.936] & \quad [3.704] & \quad [10.211] & \\ (0.361) & \quad (0.036) & \quad (0.000) & \\ a_0 = 0.213 & \quad a_1 = 0.132 & \quad a_2 = 0.12 & \\ [0.069] & \quad [1.196] & \quad [14.456] & \quad R^2 = 0.793, \\ (0.945) & \quad (0.240) & \quad (0.000) & \end{aligned}$$

где в первом ряду идут оценки коэффициентов, в квадратных скобках t -статистики, в круглых – p -значение.

Рассмотрим обобщение модели (2), когда введены дополнительные факторы. Рассмотрим такие факторы как номинальная ставка процента по депозитам и ставка процента по межбанковским кредитам. В первом случае уравнение спроса на деньги имеет вид:

$$\begin{aligned} \ln M0 = 1.774 + 0.0034 \ln w - 0.03r & \quad R^2 = 0.905 \\ [1.841] & \quad [34.116] & \quad [-1.734] \\ (0.075) & \quad (0.000) & \quad (0.092) \end{aligned}$$

$$\begin{aligned} \ln M2 = 2557 + 0.013 \ln w - 0.075r & \quad R^2 = 0.915. \\ [0.754] & \quad [44.370] & \quad [-1.497] \\ (0.456) & \quad (0.000) & \quad (0.144) \end{aligned} \quad (3)$$

Здесь w – средняя зарплата, r – процент по депозитам.

Во втором случае получим уравнение:

$$\ln M0 = 2.975 + 0.003 \ln w - 0.035 R_0, \quad R^2 = 0.921.$$

[3.017] [27.7] [-3.735]

(0.04) (0.000) (0.007)

$$\ln M2 = 4.857 + 0.0121 \ln w - 0.077 R_0, \quad R^2 = 0.917.$$

[1.384] [35.090] [-2.589]

(0.176) (0.000) (0.014)

Вопрос изучения спроса на деньги становится одним из актуальных вопросов макроэкономики.

В данной работе рассматривалось влияние трех экономических показателей – претендентов на роль переменной y в уравнении (1):

- ВВП;
- средней зарплаты;
- объема промышленного производства.

Дополнение модели (2) факторами увеличило объясняющую долю вариации (R^2).

Коэффициенты при факторах в формуле (3) – суть коэффициенты эластичности изменения показателя m , в зависимости от изменения регрессоров. В частности из формулы (3) следует, что при изменении уровня w на 1% $M2$ возрастает на 0.13%, при увеличении ставки по депозитам на 1% $M2$ уменьшится на 0.075%.

Литература

1. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. М.: Финансы и статистика, 1985.
2. Цыплаков А. А. Эконометрический анализ спроса на деньги в России. // Экономика и математические методы. №3, 1997, стр. 151-157.

ON APPROXIMATION OF THE SOLUTIONS OF STOCHASTIC EQUATIONS WITH θ -INTEGRALS

(P. Jablonski O.L.)
Byelorussian State University

Abstract. The paper investigates a problem of approximation of stochastic θ -integrals and the solutions of stochastic differential equations. It is proposed complete classifications of the ways of approximations stochastic θ -integrals and the solutions of stochastic integral equations with θ -integral in the convolution algebra.

Key words: standard process of Brownian motion, stochastic \mathcal{B} -integral, stochastic differential equation, convolution algebra, generalized random process, equations in differentials.

(*) The classical methods can't be applied to differential equations including generalized stochastic processes of the "white noise" type. For these equations, a special theory of stochastic differential equations was developed, which is based on definitions of Itô integral [1], Stratonovich integral [2], stochastic \mathcal{B} -integral [3], and others.

In papers [4, 5] the algebra of generalized stochastic processes was introduced based on the Egorov algebra [6] of generalized functions.

The main idea of the construction of algebra of generalized stochastic processes is the approximation of the stochastic processes by smooth functions and multiplying these functions. Of course, the result will depend on the ways of approximations and the right strategy is to include approximate sequence (or equivalent sequence in this or that meaning) into the definition of a generalized stochastic process. That why the problem of classifications of the ways of approximations is raised.

In this subsection we propose complete classifications of the ways of approximations stochastic \mathcal{B} -integrals in the convolution algebra (it is a sample of algebra of generalized stochastic processes). Concurrently we estimate the speed of approximations. These estimates are optimal.

Let $(\Omega, \mathcal{A}, \mathbf{P})$ be a complete probability space, $T = [0, a] \subset \mathbf{R}$, $\{\Phi_t\}_{t \in T}$ a standard flow of σ -algebras, $\Phi_a \subset \mathcal{A}$. Let $\{B(t), t \in T\}$ be a one-dimensional standard process of Φ_t -Brownian motion [7].

As a representative of generalized process of Brownian motion [5] we choose

$$B_n(t) = (B^* \rho_n)(t) = \int B(t+s) \rho_n(s) ds, \quad (1)$$

where $\rho_n(t) \in D(\mathbf{R})$, $\rho_n(t) \geq 0$, $\text{supp } \rho_n(t) \subset [0, 1/n]$ and $\int_0^{1/n} \rho_n(s) ds = 1$.

Lemma 1. For any $t \geq 0$, $h_n > 0$, $n \in \mathbf{N}$ the following equality is valid:

$$\mathbf{E}[B_n(t+h_n) - B_n(t)]^2 = h_n \iint_{\substack{0 \leq s, r \leq 1/n \\ |s-r| \leq h_n}} \left(1 - \frac{|s-r|}{h_n}\right) \rho_n(s) \rho_n(r) ds dr = h_n K(n, h_n).$$

Remark 1. It is evident, that $0 \leq K(n, h_n) \leq 1$ for any $h_n > 0$, $n \in \mathbf{N}$.

In what follows, C is an absolute constant, independent from n, h_n, t .

As a representative of the generalized function \tilde{f} [6], associating function $f: \mathbb{R} \rightarrow \mathbb{R}$, we choose

$$f_n = f * \rho_n, \quad (2)$$

where ρ_n from equality (1). The following theorems give complete classification of ways of approximations of stochastic θ -integrals, $0 \leq \theta \leq 1$ in the convolution algebra.

Theorem 1. Let $f \in C_B^2(\mathbb{R})$, $f \neq \text{const}$. The finite sum $\sum_{k=1}^m f_n(B_n(\tau_i + (k-1)h_n)) \times [B_n(\tau_i + kh_n) - B_n(\tau_i + (k-1)h_n)]$ converges in $L^2(\Omega, \mathcal{A}, \mathbb{P})$ and uniformly in $t \in T$ as $n \rightarrow \infty$, $h_n \rightarrow 0$ if, and only if numerical sequence $K(n, h_n)$ converges as $n \rightarrow \infty$, $h_n \rightarrow 0$.

Theorem 2. Let $f \in C_B^2(\mathbb{R})$, $\theta \in [0; 1/2]$, then the inequality is valid

$$\sup_{t \in T} \mathbf{E} \left[\sum_{k=1}^m f_n(B_n(\tau_i + (k-1)h_n)) [B_n(\tau_i + kh_n) - B_n(\tau_i + (k-1)h_n)] - (\theta) \int f(B(s)) dB(s) \right]^2 \leq C h_n + C/n + C(K(n, h_n) - (1 - 2\theta))^2.$$

Theorem 3. Let $f \in C_B^2(\mathbb{R})$, $f \neq \text{const}$. The finite sum $\sum_{k=1}^m f_n(B_n(\tau_i + kh_n)) \times [B_n(\tau_i + kh_n) - B_n(\tau_i + (k-1)h_n)]$ converges in $L^2(\Omega, \mathcal{A}, \mathbb{P})$ and uniformly in $t \in T$ as $n \rightarrow \infty$, $h_n \rightarrow 0$ if, and only if numerical sequence $K(n, h_n)$ converges as $n \rightarrow \infty$, $h_n \rightarrow 0$.

Theorem 4. Let $f \in C_B^2(\mathbb{R})$, $\theta \in [1/2; 1]$, then the following inequality is valid

$$\sup_{t \in T} \mathbf{E} \left[\sum_{k=1}^m f_n(B_n(\tau_i + kh_n)) [B_n(\tau_i + kh_n) - B_n(\tau_i + (k-1)h_n)] - (\theta) \int f(B(s)) dB(s) \right]^2 \leq C h_n + C/n + C(K(n, h_n) - (2\theta - 1))^2.$$

We consider equation

$$X(t) = x + (\theta) \int_0^t f(X(s)) dB(s) + \int_0^t g(X(s)) ds, \quad t \in T, \quad (3)$$

where $x \in \mathbb{R}$, $\theta \in [0; 1]$, $f \in C_B^2(\mathbb{R})$, $g \in C_B^1(\mathbb{R})$ and the stochastic integral in the right-hand side of (3) is an θ -integral.

We investigate the problem of approximation of the solution of equation (3) by the following Cauchy problem

$$\begin{cases} X_n(t+h_n) - X_n(t) = f_n(X_n(t)) [B_n(t+h_n) - B_n(t)] + g_n(X_n(t)) h_n \\ X_n(t)|_{[0, h_n]} = X_{n0}(t), \quad t \in T, \end{cases} \quad (4)$$

where "initial value" $X_{n0}(t)$ is square integrable and $\mathcal{Q}_{t, t+h_n}$ measurable for all $t \in [0; h_n)$, B_n and f_n from representations (1) and (2) respectively, $g_n = g^* \rho_n$.

Theorem 5. Suppose $\theta \in [0; 1/2]$, $f \in C_B^2(\mathbf{R})$ and $g \in C_B^1(\mathbf{R})$. Then

$$\sup_{t \in T} \mathbf{E}[X_n(t) - X(t)]^2 \leq C \sup_{t \in [0, h_n)} \mathbf{E}[X_{n0}(t) - x]^2 + C/(n^{2/3} h_n^{1/3}) + C(K(n, h_n) - (1 - 2\theta))^2,$$

where $X_n(t)$ and $X(t)$ are the solutions of the equations (5) and (3) respectively.

Theorem 6. Suppose $\theta \in [0; 1/2]$, $f \in C_B^2(\mathbf{R})$, $f \neq \text{const}$, $g \in C_B^1(\mathbf{R})$, $n^2 h_n \rightarrow \infty$ and

$$\sup_{t \in [0, h_n)} \mathbf{E}[X_{n0}(t) - x]^2 \rightarrow 0 \text{ as } n \rightarrow \infty, h_n \rightarrow 0. \text{ Then the solution of the Cauchy problem}$$

(4) $X_n(t)$ converges in $L^2(\Omega, \mathcal{A}, \mathbf{P})$ and uniformly in $t \in T$ as $n \rightarrow \infty$, $h_n \rightarrow 0$ if, and only if numerical sequence $K(n, h_n)$ converges as $n \rightarrow \infty$, $h_n \rightarrow 0$.

For approximation the solution of equation (3) when $\theta \in (1/2; 1]$ we should consider the following finite-difference equation with outstrip

$$\begin{cases} Y_n(t+h_n) - Y_n(t) = f_n(Y_n(t+h_n)) [B_n(t+h_n) - B_n(t)] - g_n(Y_n(t+h_n)) h_n \\ Y_n(t)|_{[t, h_n]} = Y_{n0}(t), \quad t \in T. \end{cases} \quad (5)$$

We use the same notation as in Cauchy problem (4).

Theorem 7. Suppose $\theta \in [1/2; 1]$, $f \in C_B^2(\mathbf{R})$ and $g \in C_B^1(\mathbf{R})$. Then

$$\sup_{t \in T} \mathbf{E}[Y_n(t) - X(t)]^2 \leq C \sup_{t \in [0, h_n)} \mathbf{E}[Y_{n0}(t) - x]^2 + C/(n^{2/3} h_n^{1/3}) + C(K(n, h_n) - (2\theta - 1))^2,$$

where $Y_n(t)$ and $X(t)$ are the solutions of the equations (5) and (3) respectively.

Theorem 8. Suppose $\theta \in [1/2; 1]$, $f \in C_B^2(\mathbf{R})$, $f \neq \text{const}$, $g \in C_B^1(\mathbf{R})$, $n^2 h_n \rightarrow \infty$ and

$$\sup_{t \in [0, h_n)} \mathbf{E}[Y_{n0}(t) - x]^2 \rightarrow 0 \text{ as } n \rightarrow \infty, h_n \rightarrow 0. \text{ Then the solution of the Cauchy problem}$$

(5) $Y_n(t)$ converges in $L^2(\Omega, \mathcal{A}, \mathbf{P})$ and uniformly in $t \in T$ as $n \rightarrow \infty$, $h_n \rightarrow 0$ if, and only if numerical sequence $K(n, h_n)$ converges as $n \rightarrow \infty$, $h_n \rightarrow 0$.

References

1. Itô K. Stochastic integral. - Proc. Imp. Acad. Tokyo, 1944, v. 20, p. 519-524.
2. Pugachev V.S., Sinitsyn I.N. Stochastic differential systems: Analysis and filtering. - Chichester: John Wiley & Sons, 1987.
3. Stratonovich R.L. A new form of writing stochastic integrals and equations. - Vestn. MGU, 1964, v.1, p. 3-12 (in Russian).

4. Lazakovich N.V., Stashulenok S.P., Yufereva I.V. Stochastic differential equations in the algebra of generalized random processes. – Differential Equations, 1995, v. 31, No. 12, p. 2056-2058.
5. Lazakovich N.V., Stashulenok S.P. An approximation of the Itô and Stratonovich integrals by elements of direct product of algebras of generalized random processes. – Theory Probab. Appl., 1997, v. 41, No. 4, p. 695-715.
6. Egorov Yu.V. On the theory of generalized functions. – Uspekhi Mat. Nauk, 1990, v. 45, p. 3-40 (in Russian).
7. Ikeda N., Watanabe S. Stochastic differential equations and diffusion processes. – Amsterdam-Oxford-New York, 1981.

ПОЛУБРАУДЕРОВЫ ОПЕРАТОРЫ И СВЯЗАННЫЕ С НИМИ СПЕКТРЫ.

Северенчук Н.Б.

Белорусский государственный университет

Аннотация. В данной работе рассматриваются полубраудеровские операторы и спектры, порожденные этими операторами, имеющие приложения в теории устойчивости и теории возмущений дифференциальных операторов.

Ключевые слова. Полубраудеровы операторы, полубраудеровы спектры, операторы взвешенного среднего.

Пусть $R(T)$ – область значений оператора $T: X \rightarrow X$, где X – банахово пространство; $N(T)$ – его ядро. Обозначим числовые характеристики линейного оператора T следующим образом: $\text{nul}(T) = \dim N(T)$; $\text{def}(T) = \text{codim} R(T) = \dim X / R(T)$; $\text{ind}(T) = \text{def}(T) - \text{nul}(T)$, $a(T)$ – подъем оператора T , т.е. наименьшее число $n \in \mathbb{N} \cup \{0\}$ такое, что $N(T^n) = N(T^{n+1})$; $d(T)$ – спуск оператора T , т.е. наименьшее число $n \in \mathbb{N} \cup \{0\}$ такое, что $R(T^n) = R(T^{n+1})$.

Оператор $T \in B(X)$ называется верхним полубраудеровым, если $T \in \{T \in B(X) : R(T) = \overline{R(T)}, \text{nul}(T) < \infty, a(T) < \infty\}$; оператор $T \in B(X)$ называется нижним полубраудеровым, если $T \in \{T \in B(X) : R(T) = \overline{R(T)}, \text{def}(T) < \infty, d(T) < \infty\}$; оператор T называется браудеровым, если он является одновременно верхним полубраудеровым и нижним полубраудеровым. Впервые название «полубраудеровы операторы» было введено R. Harte в книге [1] (определение 7.9.1).

Приведем некоторые свойства полубраудеровых операторов:

Если X – банахово пространство и $S, T \in BL(X, X)$, $ST = TS$, то

1. S, T – верхние полубраудеровы $\Leftrightarrow ST$ – верхний полубраудеров;

2. S, T – нижние полубраудеровы $\Leftrightarrow ST$ – нижний полубраудеров;
3. T – верхний полубраудеров, S – компактный $\Rightarrow T+S$ – верхний полубраудеров;
4. T – нижний полубраудеров, S – компактный $\Rightarrow T+S$ – нижний полубраудеров.

Рассмотрим подмножества комплексной плоскости \mathfrak{R} , определяемые полуредгольмовыми и полубраудеровыми характеристиками операторов $T-\lambda I$:

$$\Delta_1(T) := \{ \lambda \in \mathfrak{R} : R(T-\lambda I) = \overline{R(T-\lambda I)} \};$$

$$\Phi^+(T) := \{ \lambda \in \Delta_1(T) : \text{mul}(T-\lambda I) < \infty \}; \quad \Phi^-(T) := \{ \lambda \in \Delta_1(T) : \text{def}(T-\lambda I) < \infty \};$$

$$B^+(T) := \{ \lambda \in \Phi^+(T) : a(T) < \infty \}; \quad B^-(T) := \{ \lambda \in \Phi^-(T) : d(T) < \infty \}; \quad B(T) := B^+(T) \cap B^-(T).$$

Эти множества, согласно V. Rakoeci, определяют соответствующие спектры: $\sigma_{ab}(T) := \mathfrak{R} \setminus B^+(T)$ – Браудеров существенный аппроксимативный точечный спектр оператора T ; $\sigma_{db}(T) := \mathfrak{R} \setminus B^-(T)$ – Браудеров существенный аппроксимативный дефектный спектр оператора T ; $\sigma_{eb}(T) := \mathfrak{R} \setminus B(T)$ – Браудеров существенный спектр оператора T [2].

Основные термины и определения, а также другие существенные спектры, связанные с указанными выше множествами, рассматриваются в работах Ерovenko В.А. (см., например, [3]).

Некоторые свойства существенных спектров $\sigma_{ab}(T)$ и $\sigma_{db}(T)$ исследовались в работе J. Zemánek [4]. Эти спектры, порожденные полубраудеровыми операторами естественно также называть полубраудеровыми существенными спектрами.

Рассмотрим для примера операторы взвешенного среднего:

Пусть $A \in B(X)$, где X – банахово пространство последовательностей l_p , $1 \leq p < \infty$ или c . Зададим оператор A с помощью нижней треугольной матрицы с элементами $a_{nk} = p_k / P_n$, $k \leq n$, где $p_k \geq 0$, $p_0 > 0$, $P_n = \sum_{k=0}^n p_k$, удовлетворяющими условиями:

ям:

$$\lim_{n \rightarrow \infty} p_n / P_n = \delta, \quad 0 < \delta < 1 \quad \text{для } A \in B(X), \text{ где } X = l_p, \quad 1 \leq p < \infty \text{ или } X = c \quad (1)$$

$$\lim_{n \rightarrow \infty} n p_n / P_n = \alpha > 1/p, \quad \lim_{n \rightarrow \infty} (n p_n / P_n - (n+1) p_{n+1} / P_{n+1}) = 0 \quad \text{для } A \in B(l_p), \quad 1 < p < \infty \quad (2)$$

Так для полубраудеровых существенных спектров операторов взвешенного среднего $A \in B(l_p)$, $1 \leq p < \infty$ или $A \in B(c)$ при условии, что матрица A удовлетво-

ряет условию (1) и диагональ матрицы A не содержит бесконечное число одинаковых элементов справедливо:

$$\sigma_{ab}(A) = \{\lambda \in \mathbb{R} : |\lambda - 1/(2-\delta)| = (1-\delta)/(2-\delta)\},$$

$$\sigma_{db}(A) = \sigma_{cb}(A) = \{\lambda \in \mathbb{R} : |\lambda - 1/(2-\delta)| \leq (1-\delta)/(2-\delta)\}.$$

Автором были рассмотрены и другие случаи, связанные с изменением условий, накладываемых на оператор, а также другие существенные спектры, связанные с этими операторами.

Литература

1. Harte R. // Invertibility and Singularity for Bounded Linear Operators. Marsel Dekker, New York, 1988. 590P.
2. Rakoevi V. Semi-Browder operators and perturbations // Studia mathematica. 1997. Vol.122. №2. P.131-137.
3. Еровенко В.А. О теореме Вейля для существенных спектров Като и Фредгольма замкнутых операторов в банаховом пространстве // Вестн НАН Беларуси. Сер. физ.-мат. наук. 1998. №1. С.18-23.
4. Zemánek J. Compressions and the Weyl-Browder spectra // Proceedings of the Royal Irish Academy. 1986. Vol.86A. №1. P.57-62.

СТАТИСТИЧЕСКИЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ, ОПИСЫВАЕМЫХ AFRIMA-МОДЕЛЬЮ В ПРОСТРАНСТВЕ СОСТОЯНИЙ

Лобач И.В.

Белорусский государственный университет

г. Минск

This paper develops a state space modeling for long-range dependent data. It is shown that by using the Kalman filter, the exact likelihood function can be computed recursively in a finite number of steps.

Ключевые слова: временные ряды, фильтр Калмана-Бьюси, ARFIMA-модель.

Эмпирический анализ эволюции финансовых, экономических, социологических и многих других индексов обычно начинается с построения вероятностно-статистической модели. В общей теории временных рядов имеется целый арсенал разнообразных "стандартных" линейных моделей, среди которых в первую очередь необходимо назвать [1] такие, как MA(q), AR(p), ARMA(p, q), ARFIMA(p, q, d). Данные модели широко используются, особенно в предполо-

жении их стационарности. Причины популярности этих моделей кроются, с одной стороны в их простоте, с другой стороны, в том, что уже с небольшим числом параметров ими можно хорошо аппроксимировать весьма широкий класс стационарных последовательностей. Динамика ряда экономических показателей может быть описана процессами, имеющими "долгую" память. Так, например, в [2] исследованы результаты ежемесячной работы шведской биржи в период 1919-1995 гг. и еженедельной в период 1980-1990 гг. и было показано, что процессы, описывающие состояния биржи, имеют долгую память.

Одной из удобных моделей для описания таких процессов является ARFIMA (fractionally integrated autoregressive moving average)-модель [3], которая имеет вид:

$$\Phi(B)(1-B)^d y_t = \Theta(B)\varepsilon_t \quad (1)$$

где $|d| < 1/2$, $\{\varepsilon_t\}$ -гауссовская последовательность белого шума, B -оператор сдвига влево $B y_t = y_{t-1}$; $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$ AR-оператор, $\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$ MA-оператор. Заметим, что при $d=0$ получим ARMA-модель.

В экономике отклонения от положения равновесия обычно описываются ARIMA(p, q, d)-моделью при $d=1$. Однако данное предположение является весьма обременительным. Можно описать динамику этих отклонений ARIMA(p, q, d)-моделью при $d < 1$. Таким образом, получим ARFIMA(p, q, d)-модель. В работе [4] с помощью ARFIMA-модели была исследована средняя температура в Европе за 1944-1988 года и получены интересные результаты моделирования и прогнозирования.

В данной работе исследуются методы оценивания параметров ARFIMA-модели. Существует несколько методов оценивания. Это- R/S-алгоритм [1], байесовский метод оценивания и другие.

Известно, что метод максимального правдоподобия является одним из самых эффективных. Он обладает свойствами асимптотической нормальности, несмещенности и эффективности. Однако существуют проблемы, в частности связанные с вычислением функции правдоподобия. Логарифмическая функция правдоподобия в случае гауссовости имеет вид:

$$l_n(\theta) = -\frac{1}{2} \log \det T_n(\theta) - \frac{1}{2} Y_n' T_n^{-1}(\theta) Y_n \quad (2)$$

где $[T_n(\theta)]_{r,s=1,\dots,n}$ - ковариационная матрица вектора

$Y_n = (y_1, \dots, y_n)$, $\theta = (\phi_1, \dots, \phi_p, \theta_1, \dots, \theta_q, d, \sigma_\varepsilon)$ вектор неизвестных параметров. Как

Из формулы (2), для вычисления функции правдоподобия при 100 наблюдениях необходимо вычислять определитель и обращать матрицу размерности 100×100 , нелинейно зависящую от вектора параметров. Данные вычисления приводят к большим трудностям.

В данной работе анализируется возможность применения к оцениванию параметров ARFIMA-модели подхода, основанного на использовании описания модели в пространстве состояний и применении фильтра Калмана-Бьюси[5]. Этот подход позволяет упростить функцию правдоподобия и построить достаточно эффективный алгоритм оценивания параметров модели.

Представим ARFIMA-модель в пространстве состояний. Стандартная модель в пространстве состояний имеет вид:

$$\begin{aligned} X_{t+1} &= FX_t + H\epsilon_t \\ y_t &= GX_t + \eta_t \end{aligned} \quad (3)$$

где y_t — наблюдаемая реализация, $\{\epsilon_t\}, \{\eta_t\}$ — последовательности независимых одинаково распределенных случайных величин с нулевым математическим ожиданием и единичной дисперсией.

Непосредственной проверкой можно убедиться, что ARFIMA-модель (1) имеет вид (3), если

$$X_t = \begin{bmatrix} y(t-1) \\ y(t+1/2-1) \\ y(t+2/2-1) \\ \dots \end{bmatrix}, \quad y(t+1/2) = E(y_t | y_j, y_{j-1}, \dots),$$

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}, \quad H = [\varphi_1, \varphi_2, \dots], \quad G = [1, 0, 0, \dots]$$

Теорема. Если последовательность наблюдений $y_t, t=1, n$, порождается процессом, описываемым системой (3), то функция правдоподобия может быть представлена в виде:

$$J_n = -\frac{1}{2} \left\{ n \log 2\pi + \sum_{t=1}^n \log \Delta_t + n \log \sigma_\epsilon^2 + \frac{1}{\sigma_\epsilon^2} \sum_{t=1}^n \frac{(y_t - y_t^*)^2}{\Delta_t} \right\}, \quad (4)$$

где для $\Delta_t, y_t^*, \sigma_\epsilon^2$ получено представление, которое является очень громоздким и поэтому не приводится.

Для вычисления функции правдоподобия необходимо задать начальные данные: $X_1=0, \theta_0, E[X_1, X_1'] = (\omega_{ij}^{-1}(\theta))_{i,j=1,2,\dots,n}, \omega_{ij}^{-1}(\theta) = \sum_{k=0}^{\infty} \varphi_{i+k}(\theta)\varphi_{j+k}(\theta)$

Таким образом, получены рекуррентные соотношения для вычисления логарифмической функции правдоподобия.

Проведен численный анализ логарифмической функции правдоподобия при $y_i = (1-B)^{-d} \varepsilon_i, \sigma_\varepsilon=0.9, n=50$ и получены следующие результаты:

$$d = 0.1, \hat{d} = 0.092, d = 0.2, \hat{d} = 0.208, d = 0.3, \hat{d} = 0.308, d = 0.4, \hat{d} = 0.407.$$

d - истинное значение параметра, \hat{d} -оценка.

Результаты моделирования показывают достаточно высокую степень точности оценивания неизвестного параметра d , при увеличении объема выборки точность оценивания увеличивается.

Применение фильтра Калмана-Бьюси для анализа процессов с "долгой" памятью позволяет существенным образом упростить вид функции правдоподобия, с помощью которой проводится оценивание неизвестных параметров. Результаты статистического моделирования иллюстрируют эффективность данного подхода.

Литература

1. Ширяев А. Н. Основы стохастической финансовой математики. М.: Фазис, 1998
2. Lennart Berg, Johan Lyhagen. Short and long-run dependence in Swedish stock returns // Applied Financial Economics, Vol. 08,435-443
3. Ngai Hang Chang, Wilfredo Palma. State space modeling of long-memory processes // The Annals of Statistics 1998, Vol. 26, No 2, 719-740
4. Bloutsos A. A., Rossidis Z. V., Sahsamanoglou H. S. Arfima Modelig of Mean Temperature at the 1000/500 hPa Layer over Europe // SSE/EFI Working Paper Series in Economics and Finance, 1998
5. Браммер К., Зиффлинг Г. Фильтр Калмана-Бьюси. М.:Наука, 1982

О МЕТОДЕ СТРЕЛЬБЫ И ПАРАЛЛЕЛЬНОЙ ПРИСТРЕЛКИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ДУФФИНГА

Гайчук А.А., Мадорский В.М.

Брестский государственный университет

г.Брест, Бульвар Космонавтов, 12, ауд.615, кафедра ИиПМ, тел:23-01-65

Аннотация: Рассматриваются модификации методов стрельбы и параллельной пристрелки с добавлением демпфирующего множителя для расширения области их сходимости.

Задача Дуффинга

$$x'' + ax' + bx + cx^n = F(\sin \alpha t, \cos \alpha t), \quad (1)$$

$$x(0) = x(2\pi), \quad (2)$$

относится к сравнительно хорошо изученному классу нелинейных периодических задач. Достаточно подробно изучена задача (1), (2) при $a=0; n=3$ и в случае, если функция F линейна относительно $\sin \alpha t, \cos \alpha t$ (см., напр. [1] и приведенную там библиографию).

Для решения задачи Дуффинга применяют как проекционные, так и конечно-разностные методы. Применение метода Галеркина [2] часто требует привлечения 30-40 гармоник для получения приближенного решения с удовлетворительной точностью. Уравнение (1) может быть переписано в эквивалентном виде

$$\begin{cases} x' = y \\ y' = -ay - bx - cx^n + F(\sin \alpha t, \cos \alpha t) \end{cases} \quad (3)$$

Системе (3) может быть поставлена в соответствие следующая задача Коши

$$\frac{d\Phi}{dt} = A(t)\Phi, \quad \Phi(0) = E, \quad (4)$$

$$\text{Здесь } A = \begin{pmatrix} 0 & 1 \\ -b - cnx^{n-1} & -a \end{pmatrix} \quad (5)$$

где $\Phi(t)$ - так называемая фундаментальная матрица.

Если мультипликаторы системы (4) существенно отличны от единицы, решение методом Галеркина задачи (1) будет устойчивым [3]. В случае, если мультипликаторы системы (4) будут близки к единице, трудно рассчитывать на получение разумного решения с помощью метода Галеркина, так как мы стал-

квиваемся с некорректной задачей суммирования ряда Фурье. Результаты, полученные в [1], имеют место при специальных предположениях о параметрах задачи.

В связи с вышесказанным, представляется перспективным для решения задачи (1) использовать конечно-разностные методы с аппроксимацией производных формулами высокого порядка точности и дискретизацией всей задачи.

Получающиеся в процессе дискретизации нелинейные системы решаем с помощью следующего итерационного процесса

$$(\beta_n \|f(x_n)\|, E + \bar{f}'(x_n) f'(x)) \Delta x_n = -\bar{f}'(x_n) f(x_n),$$

$$x_{n+1} = x_n + \beta_n \Delta x_n, \beta_0 = 10^{-1} \div 10^{-4},$$

$$W_{n+1} = (1 - \beta_n) W_n + \beta_n^2 \|f(x_{n+1})\|, W_0 = \|f(x_0)\|, \beta_{n+1} = \min \left(1, \frac{W_n}{2 \|f(x_{n+1})\| \beta_n} \right).$$

Полученное в виде набора точек $(t_i, x(t_i))$ решение аппроксимируем с помощью ряда Фурье. Для того, чтобы избежать большого накопления погрешностей при суммировании ряда Фурье, применяем для аппроксимации "просеянное" по какому-нибудь правилу число точек $k \leq 20$. Если физические соображения отсутствуют, точки равномерно "просеивают". Аппроксимированное приближенное решение $\bar{x}(t)$ подставляем в (1), тем самым получая невязку на всей области изменения параметра t . Для достижения разумной точности аппроксимации, рекомендуется, чтобы шаг дискретизации был не более 0,01 (лучше порядка 0,001). В этом случае, как показывают расчеты, невязка становится порядка 0,001-0,0001.

Переходим к обсуждению другого эффективного метода решения задачи (1): метода стрельбы и параллельной пристрелки. Отличие общеизвестного метода стрельбы от предлагаемого нами метода состоит в том, что за счет параметра β_n в формуле поправки, мы достигаем расширения области сходимости, при этом β_n находим, как и выше. Параллельная пристрелка используется тогда, когда интервал интегрирования велик. Рассмотренные выше процедуры оказались весьма эффективными на классе периодических задач, связанных с задачами Дuffинга.

Численный эксперимент и его обсуждение

Рассмотрим уравнения:

$$a) x'' + x^3 = 25 \cos t;$$

$$b) x'' + x^3 = -\sin t + \sin^3 t;$$

Результаты прѳсчетов сведены в таблицу.

Метод	Уравн.	Входные данные	Результат
Стрельбы	(а)	S=0.5; N=64	Не работает
Стрельбы с параметром β_n	(а)	S=0.5; N=64	Кол.ит. = 59
Параллельной пристрелки	(б)	S=1.5; a=1; b=-1; N=64	Не работает
Пар.пристр. с пар. β_n	(б)	S=1.5; a=1; b=-1; N=64	N=256
Параллельной пристрелки	(б)	S=0.9; a=-1; b=1; N=64	Не работает
Пар.пристр. с пар. β_n	(б)	S=0.9; a=-1; b=1; N=64	N=1024

Из таблицы видно, что введение параметра β_n расширяет область сходимости.

Литература

1. Крюков Б.И. Вынужденные колебания существенно нелинейных систем. М. Машиностроение, 1984.
2. Urabe M/ Galerkin procedure for non-linear periodic systems. Arch. Rational Mech. Anal, 1965, v. 20, p. 120-152.
3. Stokes A. On the approximation of non-linear oscillations. - J. Differ. Equat., 1972, v. 12, №3, p. 535-558.

ОБ ОДНОМ НЕЛОКАЛЬНОМ СВЕРХЛИНЕЙНОМ ИТЕРАЦИОННОМ ПРОЦЕССЕ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ.

Кот А.В., Мадорский В.М.

Брестский государственный университет

г. Брест, Бульвар Космонавтов, 12, ауд. 615, кафедра ИиПМ, тел. 23-01-65

Аннотация: Приводится гибридная стратегия сочетания метода минимальных ошибок с методом построенным на основе метода типа Ньютона-Рафсона для решения нелокальных нелинейных уравнений.

Ключевые слова: метод-гибрид, сверхлинейная сходимость.

Для решения уравнения

$$f(x) = 0, \quad f(D \subset R^n \rightarrow R^n)$$

часто применяют следующий итерационный процесс [1]:

$$f'(x_n) \Delta y_n = -f(x_n);$$

$$y_n = x_n + \Delta y_n;$$

$$f'(x_n) \Delta x_n = -(f(x_n) + f(y_n)),$$

$$x_{n+1} = x_n + \Delta x_n;$$

Процесс (2)-(5) как показано в работе [1], обладает следующими привлекательными свойствами: локальной кубической сходимостью и локальной устойчивостью. Основным недостатком процесса (2)-(4) является узкая область сходимости, при этом за пределами области сходимости процесс теряет все свои привлекательные свойства.

В связи с вышесказанным, возникает мысль о возможности расширить область сходимости процесса (2)-(4) либо за счет введения демпфирующего множителя β_n , либо за счет организации метода-гибрида, который состоит из метода, «работающего» достаточно хорошо вдали от решения, который при выполнении определенных условий, переходит либо в метод (2)-(4), либо в метод типа (2)-(4) с демпфированием [2].

В этом случае соотношения (3), (5) переходят в (6), (7):

$$y_n = x_n + \beta_n \Delta y_n, \quad (6)$$

$$x_{n+1} = x_n + \beta_n \Delta x_n, \quad (7)$$

Параметр β_n определяется в [2] формулами

$$\beta_{n+1} = \min \left(1, \frac{\|f(x_n)\|}{\|f(x_{n+1})\|} \beta_n \right), \quad \beta_0 = 10^{-2} \div 10^{-4}. \quad (8)$$

Как показала практика решения существенно нелинейных задач, процесс (2), (6), (4), (7), хотя и является нелокальным, но область сходимости этого процесса не намного шире области сходимости процесса (2)-(4).

В связи с этим более привлекательным является идея создания метода-гибрида, состоящего из метода минимальных ошибок [3] или его нелокальной модификации и метода (2), (6), (4)-(7).

В разделе «Вычислительный эксперимент и его обсуждение» мы обсудим метод-гибрид.

Метод (2)-(4) или метод (2), (6), (4), (7) для своей реализации требует на каждом шаге вычислительного процесса дважды решать линейную систему, что требует порядка $\frac{4}{3}n^3$ арифметических операций (здесь n — размерность линейной системы); что примерно вдвое выше, чем в методе Ньютона. В связи с этим представляется привлекательным строить метод-гибрид на основе не демпфированного метода третьего порядка, а на основе метода типа Ньютона-Рафсона. При этом в качестве демпфирующих множителей принимаем итерационные параметры β_n , которые находим по правилу

$$\beta_{n+1} = \min \left(1, \frac{w_n}{\|f'(x_{n+1})\|} \beta_n \right), \quad \beta_0 = 0,01 + 0,0001, \\ w_{n+1} = (1 - \beta_n)w_n + \beta_{n+1} \frac{\|f'(x_{n+1})\|}{\beta_n}, \quad w_0 = \|f'(x_0)\|.$$

В качестве стартового метода применяется метод

$$\beta_0 = 10^{-2} - 10^{-4}; \quad x_{n+1} = x_n - \beta_n \frac{\|f'(x_n)\|^2 f'(x_n) f(x)}{\|f'(x_n) f(x)\|^2}, \quad \beta_n = \min \left(1, \frac{\|f'(x_{n-1})\|}{\|f'(x_n)\|} \beta_{n-1} \right)$$

Численный эксперимент и его обсуждение.

Рассматриваются две нелинейные системы

$$1. f_i(x) = x_i + \sum_{j=1}^n x_j - n - 1 = 0; \quad i = \overline{1, n-1}, \quad f_n(x) = \prod_{i=1}^n x_i - 1 = 0.$$

$$2. f_i(x) = x_i + \sum_{j=1}^n x_j - n - 1 = 0; \quad i = \overline{1, n-2}, \quad f_n(x) = \prod_{j=1}^n x_j - 1 = 0,$$

$$f_{n-1}(x) = \arctg x_1 + \arctg x_n - 2 \arctg 1 = 0.$$

Результаты расчетов сведены в таблицу.

Метод	Задача							
	№1, $x_i^0 = 10; \ f\ < 10^{-12}$				№2, $x_i^0 = 10; \ f\ < 10^{-12}$			
Размерность системы	10	20	30	40	10	20	30	40
Число итераций (метод-гибрид)	70	77	4	4	142	159	101	130
Число итераций (метод Работы [3]), $\alpha = 10^{-4}$	1557	1549	10354	685	747	625	275	159

Литература

- Ортега Дж., Рейнболдт В. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. М., Мир, 1975.
- Воронин Е.В.; Кот А.В.; Лобов С.Д.; Мадорский В.М. Решение нелинейных уравнений итерационными методами локально сходящимися с кубической скоростью. // Труды международной научной конференции SAATS-97 «Статистический и прикладной анализ временных рядов»; Брест, 1997, стр. 234-240.
- Фридман В.М. Итеративный процесс с минимальными ошибками для нелинейного операторного уравнения. ДАН СССР, т.139, стр. 1063-1066.

4. Мадорский В.М.: Нелокальные итерационные процессы без обращения. // Труды международной научной конференции SAATS-97 «Статистический и прикладной анализ временных рядов», Брест, 1997, стр.249-257.

РАЗЛОЖЕНИЕ В РЯД ФУРЬЕ ФУНКЦИИ, ЗАДАННОЙ ТАБЛИЧНО ИЛИ ГРАФИЧЕСКИ

Горегляд Д. А.

Брестский политехнический институт,

студент 3 курса электронно-механического факультета

Написана программа на Паскале, которая вычисляет коэффициенты Фурье функции, заданной таблично или графически.

Программа, коэффициенты, Фурье, функция, табличная, графическая.

Разложение функции в ряд Фурье, или гармонический анализ, оказывается нужным во многих чисто практических вопросах машиноведения, электротехники и пр. Но в этих случаях очень редко приходится непосредственно пользоваться формулами Эйлера-Фурье [1, с.563]:

для вычисления коэффициентов разложения. Дело в том, что функция, которую

$$a_0 := \frac{1}{2 \cdot \pi} \int_0^{2 \cdot \pi} f(x) dx,$$

$$a_n := \frac{1}{\pi} \int_0^{2 \cdot \pi} f(x) \cdot \cos(n \cdot x) dx,$$

$$b_n := \frac{1}{\pi} \int_0^{2 \cdot \pi} f(x) \cdot \sin(n \cdot x) dx, \quad n := 1, 2, 3, \dots$$

нужно подвергнуть гармоническому анализу, обычно задается таблицей своих значений или графиком. В этих условиях для вычисления коэффициентов Фурье нужно обратиться к приближенным методам. На практике приходится пользоваться лишь немногими первыми членами тригонометрического разложения. Коэффициенты Фурье в большинстве случаев быстро убывают, а с ними быстро падает и влияние далеких гармоник.

Обычно дается (или снимается с графика) ряд равноотстоящих ординат, т.е. ряд значений функции y , отвечающих равноотстоящим значениям аргумента x . По этим ординатам коэффициенты Фурье можно приближенно вычислить,

пользуясь известными формулами приближенного вычисления определенного интеграла. Но вычисления здесь оказываются довольно громоздкими, и для того чтобы упростить и автоматизировать их, на Паскале разработана программа, которая считает эти коэффициенты по формулам трапеций и по формулам Симпсона.

Пусть промежуток $[0, 2\pi]$ разделен на k равных частей и пусть известны ординаты

$$y_0, y_1, y_2, \dots, y_{k-1}, y_k = y_0$$

$$0, \frac{2\pi}{k}, 2 \frac{2\pi}{k}, \dots, (k-1) \frac{2\pi}{k}, 2\pi$$

отвечающие точкам деления

С учетом периодичности функции $Y_k = Y_0$ по формуле трапеций приближенно имеем [1, с.563]:

Приведем текст программы на языке Паскаль:

```
program garm_analiz;
```

$$a_0 := \frac{1}{k} \left| \frac{y_0}{2} + y_1 + y_2 + \dots + y_{k-1} + \frac{y_k}{2} \right|$$

$$a_n := \frac{2}{k} \left| y_0 + y_1 \cdot \cos\left(\frac{2 \cdot n \cdot \pi}{k}\right) + y_2 \cdot \cos\left(\frac{4 \cdot n \cdot \pi}{k}\right) + \dots + y_{k-1} \cdot \cos\left[\frac{2 \cdot n \cdot (k-1) \cdot \pi}{k}\right] \right|$$

$$b_n := \frac{2}{k} \left| y_1 \cdot \sin\left(\frac{2 \cdot n \cdot \pi}{k}\right) + y_2 \cdot \sin\left(\frac{4 \cdot n \cdot \pi}{k}\right) + \dots + y_{k-1} \cdot \sin\left[\frac{2 \cdot n \cdot (k-1) \cdot \pi}{k}\right] \right| \quad n := 1, 2, \dots$$

```
var
```

```
  y:array[0..100] of real;
```

```
  f,a0,am,bm,s:real;
```

```
  x,i,j,m,n:integer;
```

```
  BEGIN
```

```
    write('Введите количество ординат - ');readln(x);
```

```
    for i:=0 to x do begin
```

```
      write('Y',i, '=');readln(y[i]);end;
```

```

-write('Введите необходимое кол-во элем-ов - ');readln(m);
-{Вычисление коэффициентов по формуле трапеций;}
-writeln('По формулам трапеций получаем следующие значения:');
-s:=0;
-for i:=1 to x-1 do
-s:=s+y[i];
-a0:=(y[0]/2+y[x]/2+s)/x;
-writeln('A0=',a0);
-for j:=1 to m do begin
-s:=0;
-for i:=1 to x-1 do
-s:=s+y[i]*cos(6.28*i*j/x);
-am:=2*(y[0]+s)/x;
-writeln('A',j, '=',am);
-s:=0;
-for i:=1 to x-1 do
-s:=s+y[i]*sin(6.28*i*j/x);
-bm:=2*s/x;
-writeln('B',j, '=',bm);
-end;
-end.

```

Литература

1. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления. Т.3. - М.:Наука,1969.

АНАЛИЗ ВРЕМЕННЫХ РЯДОВ ГИДРОМЕТЕОЭЛЕМЕНТОВ НА ОСНОВЕ ОПТИМИЗАЦИИ АППРОКСИМИРУЮЩЕЙ ФУНКЦИИ

Валуев В.Е., Волчек А.А., Мешик О.П.
Брестский политехнический институт

В статье представлены критерии оптимизации аппроксимирующей функции при анализе временных рядов гидрометеозлементов

Ключевые слова: Синусоидальная, аппроксимация, полиномиальная, интерполяция, фурье - анализ, остаточная, дисперсия.

Рациональное природопользование возможно при всестороннем учете сложных взаимодействий природных и антропогенных процессов, в реализации

которых участвует множество климатических факторов. Известно, что используемые характеристики - производные климата, имеют большую пространственно-временную изменчивость (природные процессы протекают в динамической системе) и это обстоятельство учитывается при построении математических моделей. Временные ряды этих характеристик непрерывные или дискретные. Например, температуру воздуха за определенный период можно представить непрерывным временным рядом и искусственно трансформировать в некоторую последовательность средних суточных (декадных, месячных и других) значений, представляющих дискретный временной ряд. Атмосферные осадки на опорной сети фиксируются в дискретные моменты времени и представляют собой, в явном виде, дискретный временной ряд. При математическом моделировании, в этом случае, важно определить временной масштаб, т.е. период времени, для которого решается задача, и количество точек, подлежащих обработке. В качестве расчетного периода часто принимается год, и в расчетах используются среднесуточные величины гидрометеороэлементов, оценка значений которых, заданных на дискретном множестве точек, и экстраполяция их на всю область определения функции непрерывного аргумента возможна с использованием соответствующего математического аппарата (например, интерполяционных многочленов Лагранжа и Ньютона, сплайнов и полиномов различных степеней, средней квадратической аппроксимации, цепей Маркова, синусоидальной аппроксимации, разложения в ряд Фурье). Большинство временных рядов гидрометеорологических характеристик являются нестационарными квазислучайными последовательностями. Их среднее значение и дисперсия изменяются во времени, в зависимости от которого находится и функция распределения. Значения (X_i) временного ряда являются взаимосвязанными, между ними прослеживается четкая корреляция, постепенно затухающая в течение определенного периода времени, различного для конкретного гидрометеороэлемента. Для прогностической оценки и восстановления пропусков в рядах наблюдений можно на практике формализовать Марковские процессы различных порядков

$$X_i = \sum_{k=1}^n a_k X_{i-k} \quad (1)$$

где X_i - значение гидрометеороэлемента в момент времени (i); a_k - коэффициенты, определяемые особенностями временной структуры исследуемого ряда; n - порядок Марковского процесса; X_{i-k} - значения гидрометеороэлемента в предыдущие моменты времени ($i-k$). При исследовании характеристик климата

(рисунок) целесообразно использовать синусоидальную аппроксимацию, полиномиальную интерполяцию или Фурье-анализ, так как их временные ряды содержат периодическую составляющую. В случае наличия большого числа гармоник во временном ходе значений гидрометеорологических величин, следует использовать степенной полином вида

$$f(X) = \sum_{i=0}^n a_i \cdot X^i, \quad (2)$$

где a_i - постоянные коэффициенты; n - показатель степени полинома. Эмпирический или теоретический временной ряд можно разложить в ряд Фурье, сумма которого является функцией периода (2π)

$$f(X) = a_0 + \sum_{n=1}^{\infty} (a_n \cdot \cos nX + b_n \cdot \sin nX), \quad (3)$$

где a_0, a_n, b_n - коэффициенты Фурье, n - порядок гармоники.

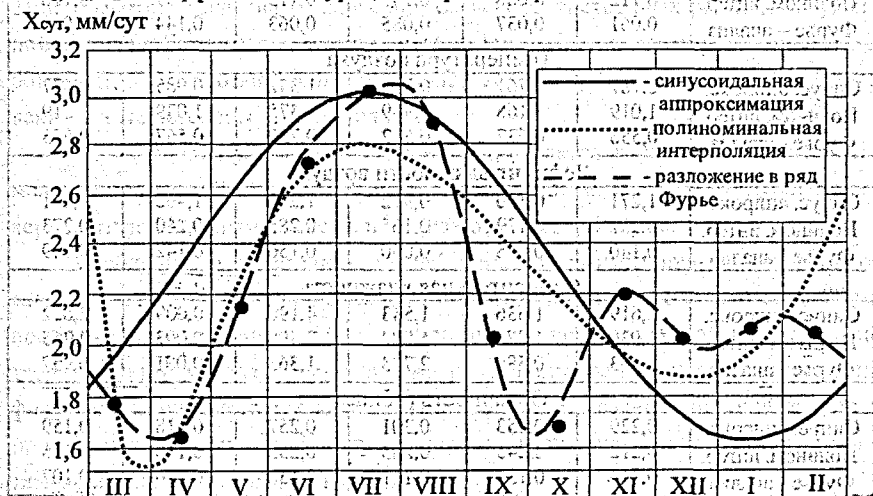


Рисунок. Среднесуточное количество атмосферных осадков в пункте Минск (точками показаны значения среднесуточного количества осадков в привязке к их суточному ходу).

При выборе способа аппроксимации для конкретного временного ряда следует руководствоваться критериями математической статистики (коэффициентом корреляции, F-критерием Фишера, остаточной дисперсией и другими). Как показали наши исследования, наиболее достоверная оценка оптимизируемой функции осуществляется по минимуму остаточной дисперсии, значения ко-

торой приведены в таблице. Из материалов таблицы видно, что описание временных рядов основных гидрометеорологических элементов предпочтительно проводить рядами Фурье. Для температур воздуха возможно использование синусоидальной аппроксимации, так как во временном ходе слабо выражены гармонические колебания эмпирических точек.

Остаточная дисперсия различных аппроксимирующих функций гидрометеорологических элементов на территории Беларуси

Таблица 1

Способ аппроксимации	Значения остаточной дисперсии					
	Минск	Брест	Гродно	Гомель	Витебск	Могилев
Атмосферные осадки						
Синус. аппрокс.	0,180	0,145	0,202	0,213	0,263	0,149
Полином. интер.	0,112	0,068	0,157	0,115	0,157	0,110
Фурье - анализ	0,061	0,037	0,085	0,063	0,144	0,060
Температура воздуха						
Синус. аппрокс.	0,787	0,761	0,808	1,342	0,985	0,677
Полином. интер.	1,019	1,168	0,939	1,375	1,039	1,219
Фурье - анализ	0,556	0,637	0,512	0,750	0,567	0,655
Дефицит влажности воздуха						
Синус. аппрокс.	1,271	0,820	0,912	1,282	1,462	1,291
Полином. интер.	0,257	0,179	0,165	0,285	0,260	0,273
Фурье - анализ	0,140	0,098	0,090	0,156	0,142	0,149
Относительная влажность						
Синус. аппрокс.	2,619	1,636	1,543	4,196	0,009	2,822
Полином. интер.	1,930	1,080	5,084	2,498	0,003	1,755
Фурье - анализ	1,053	0,589	2,773	1,363	0,001	0,957
Облачность (общая)						
Синус. аппрокс.	0,229	0,253	0,301	0,257	0,118	0,159
Полином. интер.	0,212	0,245	0,203	0,253	0,140	0,196
Фурье - анализ	0,116	0,134	0,111	0,138	0,076	0,107

Установленные аппроксимирующие функции способствуют разработке имитационных моделей природных процессов; динамика которых может экстраполироваться на любой по продолжительности период. Большинство аппроксимирующих функций апробировано нами при решении прикладных задач рационального природопользования в условиях крупномасштабных мелиораций сельскохозяйственных земель.

АВТОМАТИЗИРОВАННОЕ МОДЕЛИРОВАНИЕ ДРЕНАЖА

Валуев В.Е., Волчек А.А., Мешик О.П., Мозоль Т.Е., Омелько А.А.

Брестский политехнический институт

Брест, Республика Беларусь

В работе представлены компьютерные технологии моделирования материального дренажа на основе предлагаемого пакета программ "Дренаж"

Ключевые слова: дренаж, фильтрация, расстояние, дрена, моделирование.

Тип, расположение и размеры дренажной сети - расстояние между дренами, глубина заложения, уклоны, длины и поперечные сечения дренажных линий - должны быть таковы, чтобы они: 1) создавали и поддерживали на осушаемой площади нужный водный, воздушный, термический и пищевой режимы почвы; 2) не препятствовали бы механизации сельскохозяйственных работ на осушенной площади. Естественно, что они должны соответствовать определенным требованиям сельскохозяйственного использования осушаемой площади, ее почвенным и гидрогеологическим условиям.

Глубина дрен в настоящее время принимается в зависимости от оптимальной нормы осушения (положения зеркала подземных вод относительно поверхности земли) и изменяется в пределах от 1,0 до 1,4 метра.

Необходимые расстояния между регулируемыми дренами зависят от многих сложных факторов, наиболее точно определяются по опытным данным, приближенно - по эмпирическим зависимостям, но в проектной практике - по теоретическим методам, основанным на использовании положений теории фильтрации, учитывающих комплекс природных и хозяйственных факторов и применимым ко всякому объекту мелиорации (исключение составляют земли с тяжелыми минеральными почвами).

Алгоритм фильтрационных расчетов дренажа, включает [1]: выбор расчетной фильтрационной схемы; вычисление исходных параметров (расчетного напора, коэффициентов водоотдачи, интенсивности инфильтрационного питания); определение фильтрационных сопротивлений дрен по характеру и степени вскрытия пласта; вычисление проводимости водоносного пласта; собственно определение расстояния между дренами по стандартным программам, написанным на языке Фортран - IV для ЭВМ типа ЕС [4]. При этом, технические решения проектировщик принимает постфактум, а не в процессе необходимого диа-

лога, при оперативном поэтапном контроле и корректировке результатов расчетов.

Использование современных компьютерных технологий позволило нам реализовать задачу в среде программирования VB 5 и Access 7, стандартных программ фирмы Microsoft, иллюстративных и других возможностей ПЭВМ. Созданный пакет прикладных программ "Дренаж" используется для моделирования и оптимизации параметров горизонтального материального дренажа в условиях Белорусского Полесья. Основываясь на нормативных требованиях, мы приняли методику А.И. Мурашко, согласно которой расстояние между дренами определяется по формулам [1, 2, 3]:

$$a_d = 4 \cdot \left(\sqrt{L_{нд}^2 + \frac{H_p \cdot T}{2 \cdot q}} - L_{нд} \right), \quad (1)$$

при неустановившемся режиме фильтрации -

$$a_d = 4 \cdot \left(\sqrt{L_{нд}^2 + \frac{\tau \cdot T}{2 \cdot \mu}} - L_{нд} \right), \quad (2)$$

где $L_{нд}$ - суммарное фильтрационное сопротивление по степени и характеру вскрытия пласта; H_p - расчетный напор, м; T - проводимость пласта, м²/сут; q - интенсивность инфильтрационного питания (средний за расчетный период приток воды к дренам), м/сут; τ - время стабилизации, сут; μ - коэффициент водоотдачи. Определение междренних расстояний выполняется для двух расчетных периодов: весеннего и летне-осеннего. Весенний период является основным, его длительность 10...15 суток после окончания снеготаяния; летне-осенний - поперечный. При использовании формул, основанных на закономерностях установившейся фильтрации, в весенний период необходимо обеспечить оптимальную норму осушения к началу полевых работ и вегетации растений с выделением двух вариантов: при отсутствии затопления поверхности земли за счет удержания уровня грунтовых вод (УГВ) к началу расчетного периода на некоторой (адекватной) глубине; при полном насыщении пор почвы водой и затоплении поверхности расчетным слоем. Вариантами поперечных расчетов на летне-осенний период являются: обеспечение оптимальной нормы осушения в этот период при выпадении интенсивных дождей; обеспечение необходимой скорости освобождения пахотного слоя почвы от гравитационной воды за период времени, равный 1...3 сут

кам. При использовании формул, в основе которых заложены закономерности неустановившейся фильтрации, расчеты ведутся для условия допустимого подъема УГВ; в случае положения УГВ к началу снеготаяния на большой глубине, расчеты выполняются, исходя из недопущения подъема УГВ к началу полевых работ выше оптимальной для данного периода нормы осушения. В летне-осенний период главным условием при расчетах является недопущение подъема УГВ от интенсивных дождей, выше нормы осушения в этот период или недопущение затопления корнеобитаемого слоя почвы.

Таким образом, разработанный, апробированный и предлагаемый производственным организациям к массовому использованию программный комплекс "Дренаж" позволяет автоматизировать в едином контексте процессы решения следующих инженерных задач:

1. моделирование условий работы дренажа: а) по режиму грунтовых вод - установившийся, неустановившийся; б) по характеру изменения УГВ - заданное понижение, допустимый подъем; в) по литологическому строению - однородные, двухслойные, трехслойные грунты;
2. многовариантное моделирование конструктивных решений дрен по материалу труб и фильтров и по схеме их укладки (семнадцать вариантов);
3. прогнозирование эффективности работы дренажа в процессе его эксплуатации с учетом заиливания труб и фильтров, изменений водно - физических свойств почвогрунтов в процессе мелиорации;
4. автоматизированное проектирование элементов гидромелиоративных систем (САПР ГМС) и оснащение автоматизированного рабочего места (АРМ) проектировщика модулем "Дренаж"; принятие инженерных решений в диалоговом режиме путем многовариантной проработки, наглядного представления расчетных схем, динамического моделирования и корректировки расчетных методик с использованием материалов полевых изысканий и исследований для реальных условий ГМС;
5. обеспечение высокой производительности проектных расчетов и наглядности представления результатов, а также многократную корректировку полученных результатов в процессе принятия решений.

Литература

1. Мурашко А.И. Сельскохозяйственный дренаж в гумидной зоне. - М., 1982. - 126 с.

2. Руководство по проектированию и изысканиям объектов мелиоративного и водохозяйственного строительства в Белорусской ССР. (РПИ-82). Часть II. Осушительные и осушительно-увлажнительные системы. Книга 1. Осушительные системы самотечные. - Мн.: Белгипроводхоз, 1985. - 280 с.
3. СНиП 2.06.03-85: Мелиоративные системы и сооружения / Госстрой СССР. - М.: ЦИТП Госстроя СССР, 1986. - 60 с.
4. Установление расстояний между дренами: Дополнение №1 к Руководству по проектированию осушительных систем сельскохозяйственного назначения. - Мн.: Ураджай, 1981. - 70 с.

НЕСОВМЕСТНЫЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

Дереченник В.С.

Брестский политехнический институт

Исследованы несовместные системы линейных алгебраических уравнений; разработана программа на языке Pascal для нахождения наилучшего в смысле среднего квадратичного отклонения решения переопределенных несовместных систем по методу наименьших квадратов.

Ключевые слова: линейные уравнения; несовместная система; определитель

Грама; квадратичное отклонение;

Пусть дана некоторая несовместная система линейных уравнений вида

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{im}x_m = b_i, \quad i = 1, n \quad (1)$$

Если вместо неизвестных x_j ($j=1, m$) подставить какие-либо числа ξ_j ($j=1, m$), то мы получим результаты $\gamma_1, \gamma_2, \dots, \gamma_n$, отличные от чисел b_1, b_2, \dots, b_n .

Возникает задача: при известных значениях чисел a_{jk} и b_j ($k=1, m, j=1, n$) найти такие числа $\xi_1, \xi_2, \dots, \xi_m$, чтобы квадратичное отклонение δ^2 результатов $\gamma_1, \gamma_2, \dots, \gamma_n$ от нужных величин b_1, b_2, \dots, b_n оказалось наименьшим $\delta^2 = \sum_{j=1}^n (\gamma_j - b_j)^2$,

а также найти это минимальное отклонение.

Системы вида (1) возникают в практических задачах обработки результатов измерений, в которых коэффициенты ξ_j линейной зависимости b от величин a_j ($j=1, m$)

$$b = \xi_1 a_1 + \xi_2 a_2 + \dots + \xi_m a_m$$

должны быть найдены из результатов измерений величин a_j ($j=1, m$) и соответствующих значений b_i

$$b_i = \xi_1 a_{i1} + \xi_2 a_{i2} + \dots + \xi_m a_{im}, \quad \text{где } i = 1, n.$$

Эта система, вследствие неизбежных ошибок измерений, будет чаще всего несовместной, решение которой сводится к нахождению таких коэффициентов ξ_j , чтобы каждое уравнение удовлетворилось, хотя бы и приблизительно, но с наименьшей погрешностью.

Знание величины δ^2 в этом случае полезно тем, что помогает оценить надежность измерений.

Составляя линейную комбинацию

$$\xi_1 a_1 + \xi_2 a_2 + \dots + \xi_m a_m, \quad (2)$$

мы получим вектор $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_m\}$. Нужно определить числа $\xi_1, \xi_2, \dots, \xi_m$ так, чтобы вектор γ по норме имел наименьшее возможное отклонение от заданного вектора $b = \{b_1, b_2, \dots, b_n\}$.

Так как для переопределенных систем $m < n$, то совокупность всех линейных комбинаций векторов a_1, a_2, \dots, a_m образует подпространство $L = L(a_1, a_2, \dots, a_m)$. В этом подпространстве наименьшее расстояние до вектора b имеет проекция этого вектора на подпространство L . Следовательно, числа $\xi_1, \xi_2, \dots, \xi_m$ должны быть выбраны так, чтобы линейная комбинация (2) привела к проекции вектора b на подпространство.

Подчинив вектор $b - g = h$ условию ортогональности со всеми векторами a_1, a_2, \dots, a_m , получим систему вида

$$\begin{cases} (a_1, a_1)\xi_1 + (a_2, a_1)\xi_2 + \dots + (a_m, a_1)\xi_m = (b, a_1), \\ (a_1, a_2)\xi_1 + (a_2, a_2)\xi_2 + \dots + (a_m, a_2)\xi_m = (b, a_2), \\ \dots \\ (a_1, a_m)\xi_1 + (a_2, a_m)\xi_2 + \dots + (a_m, a_m)\xi_m = (b, a_m). \end{cases} \quad (3)$$

Главный определитель системы (3) — это определитель Грама $D = G(a_1, a_2, \dots, a_m)$ [1]. Так как векторы образуют базис (линейно-независимый), то $D \neq 0$. Следовательно, система (3) имеет единственное решение:

$$\xi_j = 1/D \begin{vmatrix} (a_1, a_1) & \dots & (a_{j-1}, a_1) & (b, a_1) & (a_{j+1}, a_1) & \dots & (a_m, a_1) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ (a_1, a_m) & \dots & (a_{j-1}, a_m) & (b, a_m) & (a_{j+1}, a_m) & \dots & (a_m, a_m) \end{vmatrix} \quad (4)$$

Оценим и само отклонение δ . Величина δ = высота $(m+1)$ -мерного параллелепипеда, построенного на векторах a_1, a_2, \dots, a_m, b , и равна

$$\delta = \frac{V(a_1, a_2, \dots, a_m, b)}{V(a_1, a_2, \dots, a_m)}$$

$$\delta^2 = \frac{G(a_1, a_2, \dots, a_m, b)}{G(a_1, a_2, \dots, a_m)} \quad (5)$$

На примере несовместной системы

$$\begin{cases} 3x_1 + 2x_2 + x_3 = 5, \\ 2x_1 + 3x_2 + x_3 = 1, \\ 2x_1 + x_2 + 3x_3 = 7, \\ x_1 + 2x_2 + 4x_3 = 0, \end{cases}$$

показано, что квадратичное уклонение δ^2 действительно является наименьшим. Также составлена программа на языке PASCAL, которая позволяет находить решение переопределенной несовместной системы с любым числом уравнений, превышающим число неизвестных, по методу наименьших квадратов, рассмотренному выше.

Литература

1. Шиллов Г. Е. Математический анализ. Конечномерные линейные пространства. - М.: Наука, 1969.

АСФЕРИЧЕСКИЕ БАЗИСЫ СООТНОШЕНИЙ ПРО- p -ГРУПП

Шишкевич А. А.

Белорусский государственный университет

г. Минск, пр. Скорины-4, БГУ, ММФ, кафедра Высшей Алгебры

Пусть p — произвольное простое число. Все группы, встречающиеся в статье, предполагаются про- p -группами, подгруппы — замкнутыми; все модули предполагаются компактными модулями над пополненным групповым кольцом ΩG , где Ω — поле из p элементов, G — соответствующая про- p -группа; все морфизмы предполагаются непрерывными.

Следующее определение класса асферических про- p -групп и приведённые ниже факты принадлежат О. В. Мельникову. Пусть

$$1 \longrightarrow N \longrightarrow F \xrightarrow{\varphi} G \longrightarrow 1 \quad (1)$$

некоторое копредставление про- p -группы G . Группа G действует слева (с помощью сопряжения) на абелевой группе экспоненты p $\bar{N} := N/N^*$, $N^* = N^p[N, N]$ — группа Фраттини N . ΩG -модуль \bar{N} называется модулем соотношений группы G . По определению, группа G асферична, если существует копредставление (1), модуль соотношений \bar{N} которого изоморфен некото-

рому пермутационному модулю $\Omega(T)$ (здесь T — некоторое проконечное G -пространство). Асферичность про- p -группы G не зависит от выбора копредставления, а именно, если асферично какое-нибудь из копредставлений, то асферично и любое другое. Стабилизаторы элементов G -пространства T суть конечные циклические p -группы, тривиально пересекающиеся между собой. Подгруппы асферических групп также асферичны.

Если асферическая группа G конечно определена, то найдётся набор соотношений $\{r_1, \dots, r_m\} \subseteq N$, порождающих нормальный делитель N , для которых $\bar{N} \cong \bigoplus_{i=1}^m \Omega G \cdot r_i$, причём левый аннулятор элемента r_i в кольце ΩG совпадает с ядром $I_G(\varphi(S_i))$ канонического эпиморфизма $\Omega G \rightarrow \Omega(G/\varphi(S_i))$, где S_i — централизатор элемента $r_i \in F_i$ (циклическая группа, порождённая $\sqrt{r_i}$ — максимальным корнем из r_i). Тогда $\bar{N} \cong \bigoplus_{i=1}^m \Omega(G/\varphi(S_i)) = \Omega(UG/\varphi(S_i))$ — конечно порождённый пермутационный модуль. Существование такой системы соотношений отражает неформальное содержание понятия асферичности — отсутствие каких-либо дополнительных соотношений между соотношениями, кроме естественных. В случае, когда множество определяющих соотношений недискретно, ситуация несколько сложнее.

Асферическим базисом соотношений копредставления (1) про- p -группы G назовём замкнутое подмножество $R \subseteq_C N$, такое, что $N = (R)^F$ и $\bar{N} \cong \bigoplus_{r \in R} \Omega G \cdot r$, $\text{Ann}_{\Omega G}(\bar{r}) = I_G(\varphi(S_r))$. Подобное определение нуждается в непростой проверке существования прямой суммы стоящего справа пучка одно-порождённых G -модулей. (По поводу прямых сумм пучков модулей см. Мельников [1].) Понятно, что из существования асферического базиса соотношений вытекает асферичность копредставления (1): $\bar{N} \cong \Omega(T)$, где T — образ непрерывного отображения $\mu: G \times R \rightarrow \bar{N}$, $(g, r) \mapsto g \cdot r$.

В данной заметке мы сформулируем критерий асферичности базиса соотношений, аналогичный полученному для абстрактных групп (см. Чисвели, Коллинз, Хюбшманн [2]) и получим его простейшие следствия.

Придерживаясь выбранных выше обозначений, рассмотрим свободную F -операторную про- p -группу \tilde{N} с базисом R . Ясно, что $\tilde{N} = F(F \times R)$ — свободная про- p -группа над компактным пространством $F \times R$. Имеет место канонический эпиморфизм $\theta: \tilde{N} \rightarrow N$, $(u, r) \mapsto uru^{-1}$ — «вычисляющая» функция. Ядро $E = \text{Ker } \theta$ будем называть множеством тождеств копредставления (1). Элементарными тождествами Пайфер назовём семейство элементов \tilde{N} $P_0 = \{ab^{(\theta(b)^{-1}a)^{-1}b^{-1}}, aba^{-1}(\theta(a)b)^{-1}, cd^{-1}\}$, где $a, b, c, d \in F \times R$, $pr_Rc = pr_Rd$, $\theta(c) = \theta(d)$. Множеством тождеств Пайфер назовём нормальное замыкание элементарных тождеств Пайфер в \tilde{N} , $P = (P_0)^{\tilde{N}} \triangleleft \tilde{N}$. Ясно, что $P \subseteq E$.

Предложение. Пусть $R \subseteq_c N$. В принятых обозначениях,

$E \equiv P \Leftrightarrow R$ — асферический базис соотношений.

Доказательство. Прежде всего установим следующий изоморфизм G -модулей (F -операторных групп, с тривиальным действием N):

$$\eta: \tilde{N}/P\tilde{N}^* \rightarrow \bigoplus_{r \in R} \Omega(G/\varphi(S_r)), \quad (1, r) \mapsto 1\varphi(S_r).$$

Достаточно проверить, что

$$P\tilde{N}^* = \left(n(1, r) = (1, r), {}^r(1, r) = (1, r) \mid n \in N, r \in R, s_r \in S_r \right) \tilde{N}^*. \quad (2)$$

Проверим включение слева направо. По модулю \tilde{N}^*

$$ab^{(\theta(b)^{-1}a)^{-1}b^{-1}} \equiv a^{(\theta(b)^{-1}a)^{-1}} = [a \equiv (f; r)] \equiv (1, r)^{(\theta(b)^{-1}f(1, r))^{-1}} \equiv$$

$$(1, r)^{f((1, r)(f^{-1}\theta(b)^{-1}f(1, r))^{-1})} \equiv 1, \text{ так как } f^{-1}\theta(b)^{-1}f \in N.$$

Если $\theta(f_1, r) = \theta(f_2, r)$, то $f_2^{-1}f_1 \in S_r$. Поэтому $(f_1, r)(f_2, r)^{-1}$ принадлежит правой части (2). Для тождества $aba^{-1}(\theta(a)b)^{-1}$ проверка аналогична.

Проверим включение справа налево. Имеем

$$n(1, r)(1, r)^{-1} = [n \equiv \theta(a), a \in \tilde{N}] \equiv_{\theta(a)} (1, r)(1, r)^{-1} \equiv_{\theta(a)} a(1, r)a^{-1}(1, r)^{-1} \equiv_{\theta(a)} 1 \pmod{P\tilde{N}^*} \equiv_{\theta(a)} 1 \pmod{\tilde{N}^*}.$$

Для тождества ${}^r(1, r)(1, r)^{-1}$ всё очевидно.

Рассмотрим сейчас эпиморфизм

$$\lambda: \bigoplus_{r \in R} \Omega(G/\varphi(S_r)) \xrightarrow{\eta^{-1}} \tilde{N}/P\tilde{N}^* \rightarrow \tilde{N}/E\tilde{N}^* = \bar{N}, \quad 1\varphi(S_r) \mapsto \bar{r}.$$

Ясно, что R является базисом асферичности копредставления (1), в том и только в том случае, когда λ будет изоморфизмом, то есть тогда и только то-

гда, когда $P\tilde{N}^* = E\tilde{N}^*$. Следовательно осталось убедиться в том, что $P = E$ тогда и только тогда, когда $P\tilde{N}^* = E\tilde{N}^*$. Для этого рассмотрим расширение группы E/P с помощью свободной группы N :

$$1 \rightarrow E/P \rightarrow \tilde{N}/P \rightarrow \tilde{N}/E = N \rightarrow 1.$$

Тождества Пайфер показывают, что это расширение центрально. Значит, $\tilde{N}/P = E/P \times \tilde{N}/E$. Переходя к фактору по подгруппе Фраттини, получаем $\overline{E/P} \times \overline{N} = \tilde{N}/P\tilde{N}^* = [P\tilde{N}^* = E\tilde{N}^*] = \tilde{N}/E\tilde{N}^* = N/N^* = \overline{N} \Rightarrow \overline{E/P} = 0 \Rightarrow E = P$:

Импликация из $P = E$ в $P\tilde{N}^* = E\tilde{N}^*$ очевидна. Предложение доказано.

Следствие 1.) Если $\chi: R \rightarrow F$ некоторая непрерывная функция, то множество $R_\chi = \{\chi(r)r\chi(r)^{-1} \mid r \in R\}$ — замкнуто в F . Тогда R_χ — асферический базис соотношений копредставления $G = F/(R_\chi)^F$.

Следствие 2.) Если $\alpha \in \text{Aut}(F)$, то $\alpha(R)$ — асферический базис соотношений копредставления $G = F/(\alpha(R))^F$.

Доказательство следствия вытекает из того, что

$$P_R = E_R \Rightarrow P_{R_\chi} = E_{R_\chi}, P_{\alpha(R)} = E_{\alpha(R)}.$$

Литература.

1. Мельников О.В. Подгруппы и гомологии свободных произведений проконечных групп. // Изв. АН СССР, сер. матем. — 1989. — т.53, №1. С.97-120.
2. Chiswell I.M., Collins D.J., Huebschmann J. Aspherical group presentations // Math. Z. — 1981. — Bd 178. — S. 1-36.

СОДЕРЖАНИЕ

Бахтизин В.В., Перегуд С.В. ИНТЕГРАЦИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ	3
Соколов С.А. ДИЗАЙНЕРЫ SQL ЗАПРОСОВ. КЛАССИФИКАЦИЯ. ПРИНЦИП ФУНКЦИОНИРОВАНИЯ.	6
Соколов С.А. АЛГОРИТМИЧЕСКАЯ МУЛЬТИЯЗЫЧНОСТЬ. ПРОБЛЕМЫ: ПУТИ РЕАЛИЗАЦИИ.	9
Галерштейн В.Е., Ревотюк М.П. ОРГАНИЗАЦИЯ ГРУППОВЫХ ЗАПРОСОВ К МНОГОТАБЛИЧНЫМ ДАННЫМ.....	12
Молочко Д. А., Муха В. С. ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ РАБОТЫ С ИЗОБРАЖЕНИЯМИ	15
Валиев Д.А. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ АНАЛИЗА ПРОИЗВОДСТВЕННО-ХОЗЯЙСТВЕННОЙ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ И ФИНАНСОВОГО ПЛАНИРОВАНИЯ.....	20
Валиев М.А., Железко Б.А. ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ БИЗНЕС-ПЛАНИРОВАНИЯ	23
Железко Б.А., Морозов П.Е., Неправский А.А. СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ФОНДОВОЙ БИРЖИ «ОМЕГА».....	26
Лавренова Н.М. ИСПОЛЬЗОВАНИЕ IDEFO-МЕТОДОЛОГИИ ПРИ РАЗРАБОТКЕ БАЗ ДАННЫХ.....	29
Зеневич Д.В. ОЦЕНИВАНИЕ НАЧАЛЬНЫХ ЗНАЧЕНИЙ АВТОРЕГРЕССИОННЫХ ВРЕМЕННЫХ РЯДОВ	32
Бахтизин В.В., Круглов Д.Г. КОНСОЛЬНОЕ УПРАВЛЕНИЕ РЕСУРСАМИ В ЗАДАЧАХ ПОСТРОЕНИЯ ТРЕХМЕРНОГО ИЗОБРАЖЕНИЯ.....	36
Карасик Е. А. ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	39
Тиунчик Д.В., Лобов С.Д. СИСТЕМА КОНТРОЛЯ ЗНАНИЙ.....	42
Бортник Е.М., Ревотюк М.П. КОНСТРУКТИВНАЯ КОМПОЗИЦИЯ СТРУКТУР СЕТЕВЫХ МОДЕЛЕЙ ОРГАНИЗАЦИОННО-ТЕХНОЛОГИЧЕСКИХ СИСТЕМ	46
Ревотюк М.П., Тихомирова Е.В. ПОИСК И ПРЕДСТАВЛЕНИЕ ДОПУСТИМЫХ ТЕХНОЛОГИЧЕСКИХ МАРШРУТОВ В УСТАНОВКАХ КОМПАУНДИРОВАНИЯ.....	49
Тихомирова Е.В. АНАЛИЗ ПОТЕНЦИАЛЬНОЙ ЭФФЕКТИВНОСТИ КООПЕРАТИВНЫХ СХЕМ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ	52

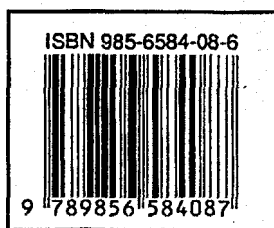
Бахтизин В.В., Крапивин В.А. ПРИМЕНЕНИЕ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ОБРАЗОВ В СРЕДСТВАХ ЗАЩИТЫ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА	55
Игнатюк О.Н. САМООБУЧАЮЩАЯСЯ НЕЙРОННАЯ СИСТЕМА ДЛЯ АВТОНОМНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ	59
Абрамук В.М., Маньяков Н.В., Махмист Л.П. СЛОВА СМЕЖНОГО КЛАССА КОДА ХЭММИНГА	64
Гладкий И.И. ВЫБОР АДАПТИВНОГО ШАГА ОБУЧЕНИЯ ПРИ ГРУППОВОМ ОБУЧЕНИИ ДЛЯ ЛИНЕЙНОЙ НЕЙРОННОЙ СЕТИ	67
Костюк Д.А. МОДЕЛИРОВАНИЕ РАСПРОСТРАНЕНИЯ ПРОДОЛЬНОГО ЗВУКА В СЛОИСТОЙ СТРУКТУРЕ, СОДЕРЖАЩЕЙ СЛОЙ ДИССИПАТИВНОЙ СРЕДЫ	69
Акулич Я.А., Лукша В.В., Шведовский П.В. ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ РЕГИОНАЛЬНЫХ АГРОЭКОЛОГИЧЕСКИХ СИСТЕМ	72
Лукша В.В., Акулич Я.А., Шведовский П.В. ОСОБЕННОСТИ ОПТИМИЗАЦИИ СТРУКТУРЫ И МОДЕЛИРОВАНИЯ ГЕНЕТИЧЕСКОЙ ЭВОЛЮЦИИ ГЕО-, ЭКО- И АГРОЭКОЛОГИЧЕСКИХ СИСТЕМ	75
Яцкевич О.С., Соколюк Н.Л., Говорова Е.И., Дыдышко М.И. УСПЕШАЕМОСТЬ СТУДЕНТОВ И ОПРЕДЕЛЯЮЩИЕ ЕЁ ФАКТОРЫ.	79
Никулин Ю.В. О РАДИУСЕ УСТОЙЧИВОСТИ ЭФФЕКТИВНОГО РЕШЕНИЯ ВЕКТОРНОЙ ЗАДАЧИ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	82
Лазуренко Ю.Н. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ НАПРЯЖЕННОГО СОСТОЯНИЯ В МАССИВЕ В ОКРЕСТНОСТИ ВЫРАБОТОК БОЛЬШИХ ПРОТЯЖЕННОСТЕЙ.	85
Стагурова О.В. ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ НАПРЯЖЕННОГО СОСТОЯНИЯ В ОБЛАСТЯХ ПОРОДНОГО МАССИВА С ГЕОЛОГИЧЕСКИМИ НАРУШЕНИЯМИ	89
Народецкий Д.А. МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ ДЕНЕЖНОГО РЫНКА НА ОСНОВЕ УРАВНЕНИЯ МАРШАЛЛА-ПИГУ	92
Jablonski O.L. ON APPROXIMATION OF THE SOLUTIONS OF STOCHASTIC EQUATIONS WITH ϕ -INTEGRALS	94
Лобач И.В. СТАТИСТИЧЕСКИЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ, ОПИСЫВАЕМЫХ AFRIMA- МОДЕЛЬЮ В ПРОСТРАНСТВЕ СОСТОЯНИЙ	100
Гайчук А.А., Мадорский В.М. О МЕТОДЕ СТРЕЛЬБЫ И ПАРАЛЛЕЛЬНОЙ ПРИСТРЕЛКИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ДУФФИНГА	104

Кот А.В., Мадорский В.М. ОБ ОДНОМ НЕЛОКАЛЬНОМ СВЕРХЛИНЕЙНОМ ИТЕРАЦИОННОМ ПРОЦЕССЕ ДЛЯ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ.....	106
Горегляд Д.А. РАЗЛОЖЕНИЕ В РЯД ФУРЬЕ ФУНКЦИИ, ЗАДАННОЙ ТАБЛИЧНО ИЛИ ГРАФИЧЕСКИ.....	109
Валуев В.Е., Волчек А.А., Мешик О.П. АНАЛИЗ ВРЕМЕННЫХ РЯДОВ ГИДРОМЕТЕОЭЛЕМЕНТОВ НА ОСНОВЕ ОПТИМИЗАЦИИ АППРОКСИМИРУЮЩЕЙ ФУНКЦИИ.....	111
Валуев В.Е., Волчек А.А., Мешик О.П., Мозоль Т.Е., Омелько А.А. АВТОМАТИЗИРОВАННОЕ МОДЕЛИРОВАНИЕ ДРЕНАЖА.....	115
Дереченник В.С. НЕСОВМЕСТНЫЕ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ.....	118
Шишкевич А.А. АСФЕРИЧЕСКИЕ БАЗИСЫ СООТНОШЕНИЙ ПРО-Р-ГРУПП.....	120

Научное издание

«СОВРЕМЕННЫЕ ПРОБЛЕМЫ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»

Труды региональной конференции молодых ученых и студентов
16-18 ноября 1999 г.



Ответственный за выпуск: Кудинов Н.В.

Редактор: Строкач Т.В.

Лицензия №882 от 30.04.99

Подписано к печати 3.01.00 Формат 60x84 1/16 Бумага писч. Гарнитура Т.Н.Р
Усл. п.л. 7,2 Уч. изд. л. 7,75 Тираж 100 экз Заказ № 15 Бесплатно. Отпечатано на
ризографе Брестского политехнического института. 224017, Брест, ул. Москов-
ская, 267.