

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

11(219)
2014

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

Кухаренко Б. Г., Солнцева М. О. Покомпонентный дискриминантный анализ результатов кластеризации многомерных траекторий 3

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

Баранов М. А., Зеленко Г. В., Рошин А. В. Проблемы маршрутизации сообщений и сенсорной локальной сети 8

Поршнев С. В., Божалкин Д. А. Технология семантического анализа дампа трафика информационных потоков в компьютерных сетях 12

Велиев Ф. М. Оптимизация вегетационного цикла роста растений по информационному критерию при использовании индекса листевой площади 19

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

Якимов Н. М., Чувашев С. Н. Программное средство для комплексного математического моделирования сложных технических объектов 23

Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Соловьев А. Н., Мячиков М. В. Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем 30

Усков А. А., Жукова А. Г. Матричное представление иерархических визуальных моделей для CASE-средств 36

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

Игнатова И. Г., Береза Е. О. Модель интеграции гетерогенных информационных ресурсов с гибким управлением связями 40

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ

Михайлов Д. М., Стариковский А. В., Конев В. Н., Смирнов А. С., Боручинкин А. Ю., Харьков С. М. Программное обеспечение для автоматизированного анализа изображений желудочно-кишечного тракта 46

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Богодистова Е. С., Мамченко А. Е., Шамров М. И. Программируемые контроллеры в компьютерных и управляющих системах 53

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

Богданов М. Р., Захаров А. В., Габидуллин Ю. З., Думчиков А. А., Дмитриев В. Г. Кодирование текста с помощью музыки 60

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

Барский А. Б. Рейтинговые системы на логических нейронных сетях 66

Аникин В. И., Карманова А. А. Обучение искусственной нейронной сети Кохона на клеточным автоматом 73

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.

ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.

КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.

ПОПКОВ Ю. С.,
чл.-корр. РАН, д. т. н., проф.

РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.

РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.

СОЙФЕР В. А.,
чл.-корр. РАН, д. т. н., проф.

СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.

СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,

чл.-корр. РАН, д. т. н., проф.

ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.

ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.

ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.

БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В.А., д. ф.-м. н., проф.

ГАЛУШКИН А.И., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.

ДОМРАЧЕВ В.Г., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.

ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.

КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.

КУРЕЙЧИК В. М., д. т. н., проф.
КУХАРЕНКО Б. Г., к. ф.-м. н., доц.

ЛЬВОВИЧ Я. Е., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.

НЕЧАЕВ В. В., к. т. н., проф.
РЯБОВ Г. Г., чл.-корр. РАН, д. т. н., проф.

СОКОЛОВ Б. В., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)

ФОМИЧЕВ В. А., д. т. н., проф.
ЧЕРМОШЕНЦЕВ С. Ф., д. т. н., проф.

ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.

ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

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

CONTENTS

INTELLIGENT SYSTEMS AND TECHNOLOGIES

Kukhareno B. G., Solntseva M. O. Component Wise Discriminant Analysis of Multi Dimensional Trajectory Clustering 3

COMPUTING SYSTEMS AND NETWORKS

Baranov M. A., Zelenko G. V., Roschin A. V. Message Routing Problems in Sensory LAN 8

Porshnev S. V., Bozhalkin D. A. Semantic Analysis Technology of Dataflow in Computer Networks Traffic Dump 12

Veliyev F. M. Optimization of Development Stage of Vegetation on Information Criterion Using the Leaf Area Index 19

MODELING AND OPTIMIZATION

Yakimov N. M., Chuvashov S. N. Application Software for Composite Mathematical Modeling of Complex Technical Objects 23

Stempkovskiy A. L., Telpukhov D. V., Solovyev R. A., Solovyev A. N., Myachikov M. V. Fault Simulation Technique for Logic Circuits Reliability Characteristics Evaluation 30

Uskov A. A., Zhukova A. G. Matrix Representation Hierarchical Visual Models for GASE-Tools 36

INFORMATION TECHNOLOGIES IN EDUCATION

Ignatova I. G., Bereza E. O. The Model of Heterogeneous Information Resources Integration with Flexible Communication Management 40

INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

Mikhaylov D. M., Starikovskiy A. V., Konev V. N., Smirnov A. S., Boruchinkin A. Yu., Kharkov S. M. Software for the Automated Analysis of the Gastrointestinal Tract Images 46

APPLIED INFORMATION TECHNOLOGIES

Bogodistova E. S., Mamchenko A. E., Schamrov M. I. Programmable Controllers in Computer and Control Systems 53

CRYPTOSAFETY INFORMATION

Bogdanov M. R., Zakharov A. V., Gabidullin Ju. Z., Dumchikov A. A., Dmitriev V. G. Coding Text with Music 60

Journal-in-journal

NFUROTECHNOLOGIES

Barky A. B. The Rating Systems on Base of the Logical Neural Networks 66

Anikin V. I., Karmanova A. A. Learning of Kohonen Artificial Neural Network by Help of Cellular Automaton 73

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.

Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.

Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.

Popkov Yu. S., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Soifer V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.

Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.

Chaplygin Yu. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.

Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.

Barsky A. B., Dr. Sci. (Tech.), Prof.

Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Galushkin A. I., Dr. Sci. (Tech.), Prof.

Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.

Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.

Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)

Kulagin V. P., Dr. Sci. (Tech.), Prof.

Kureichik V. M., Dr. Sci. (Tech.), Prof.

Kukhareno B. G., Cand. Sci. (Phys.-Math.)
Ljovich Ya. E., Dr. Sci. (Tech.), Prof.

Mikhailov B. M., Dr. Sci. (Tech.), Prof.

Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Sokolov B. V., Dr. Sci. (Tech.)

Uskov V. L. (USA), Dr. Sci. (Tech.)

Fomichev V. A., Dr. Sci. (Tech.), Prof.

Chermoshentsev S. F., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.

Grigorin-Ryabova E. V.

Lysenko A. V.

Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

УДК 519.233

Б. Г. Кухаренко, канд. физ.-мат. наук, ст. науч. сотр., вед. науч. сотр.,
Институт машиноведения имени А. А. Благонравова РАН, г. Москва,
e-mail: kukharenko@imash.ru

М. О. Солнцева, аспирант, Московский физико-технический институт (ГУ),
e-mail: solnceva.chalei@gmail.com

Покомпонентный дискриминантный анализ результатов кластеризации многомерных траекторий

Многомерный дискриминантный анализ применяется к результатам кластеризации пространственных траекторий объектов на основе метода полиномиальных регрессий. Преимуществом многомерного дискриминантного анализа является сокращение размерности исходных данных в подпространствах, представляющих кластеры. С помощью многомерного дискриминантного анализа одна из проекций на координатные оси в исходном пространстве траекторий выделяется как наиболее информативная для определения структуры рассматриваемых кластеров.

Ключевые слова: анализ данных, многомерные траектории, кластеризация, модель гауссовых смесей, дискриминантный анализ

Введение

Траектории движения объектов представляются кривыми в трехмерном пространстве. Естественным критерием кластеризации пространственных кривых является визуальное сходство их геометрической формы. При этом траектории, принадлежащие одному кластеру, могут быть разнесены в пространстве. Для кластеризации траекторий используют различные методы, из которых наиболее эффективным является метод полиномиальных регрессий [1, 2], так как он учитывает пространственную форму этих кривых.

Поскольку траектории объектов представляются большим числом точек, то для упрощения кластеризации траекторий также используется подход, основанный на отображении в различные, в том числе и многомерные, пространства характеристик траекторий [3, 4]. Однако этот подход демонстрирует некоторую нестабильность результатов. Результаты кластеризации на основе подхода, описанного в работе [3], улучшаются после нескольких итераций [4].

В трехмерном пространстве траектории объектов представляются в виде $T_j = \{(x_j[i], y_j[i], z_j[i]), i = \overline{1, N_j}\}$, $j = \overline{1, J}$, где декартовы координаты $(x_j[i], y_j[i], z_j[i])$ соответствуют оцениваемому положению j -го объекта. В настоящей работе исследуется значимость различных проекций простран-

ственных траекторий на оси координат для кластеризации этих траекторий. Траектории T_j , $j = \overline{1, J}$ описываются последовательностями значений координат $X_j = \{x_j[i], i = \overline{1, N_j}\}$, $Y_j = \{y_j[i], i = \overline{1, N_j}\}$ и $Z_j = \{z_j[i], i = \overline{1, N_j}\}$, называемых компонентами. Различные компоненты неодинаково отражают информацию о принадлежности рассматриваемых траекторий к имеющимся кластерам.

Информативность компонент, т. е. степень сохранности информации о принадлежности пространственных траекторий к кластерам, анализируется отдельно для каждой компоненты посредством многомерного дискриминантного анализа [5, 7]. В численном эксперименте используются результаты совместной кластеризации траекторий движения самолетов в окрестности международного аэропорта г. Сан-Франциско, полученные с помощью метода полиномиальных регрессий [2] по данным, находящимся в свободном доступе на сайте <http://c3.nasa.gov/dashlink/resources/132/>. Эта кластеризация далее называется *совместной априорной кластеризацией* и используется для оценки эффективности многомерного дискриминантного анализа.

Для каждой из компонент траектории, кластеризованные методом полиномиальных регрессий [2], разделяются на две равночисленные выборки. Та

выборка, для которой распределение траекторий по кластерам оказывается более неоднородным, используется в качестве *обучающей* выборки для построения модели структуры кластеров. Вторая выборка является *тестируемой*. В результате применения многомерного дискриминантного анализа [5, 7] к обучающей выборке вычисляются параметры модели гауссовой смеси. Эти параметризованные модели, полученные для каждой компоненты траекторий, используются для предсказания распределения траекторий тестируемой выборки по кластерам.

Информативность кластеризации по каждой компоненте траекторий оценивается следующим образом. Распределение по кластерам траекторий из тестируемой выборки сравнивается с распределением этих же траекторий по кластерам в совместной априорной кластеризации. Доля одинаково кластеризованных траекторий рассматривается как индекс относительной информативности компоненты. Таким образом, покомпонентный анализ позволяет выделить наиболее информативную компоненту при выявлении структуры кластеров в исходном трехмерном пространстве.

Многомерный дискриминантный анализ

Кластеризация траекторий из обучающей выборки выполняется в рамках вероятностной модели, при этом каждому кластеру соответствует его распределение вероятности. Поэтому траектории из обучающей выборки, для которых кластеры являются скрытыми, описываются конечной моделью смеси распределений. Подробное описание таких моделей содержится в работе [5]. Однако методы кластеризации на основе модели смеси распределений в пространствах большой размерности, представляющих траектории, оказываются высокопараметризованными. Поэтому на практике предварительно уменьшают размерность данных траекторий с помощью метода главных компонент (*Principal Component Analysis — PCA*) [11] или методов отбора характеристик [12]. Однако уменьшение размерности данных траекторий перед кластеризацией приводит к субоптимальному представлению данных, поскольку частичная потеря информации может влиять на точность определения кластеров. Сохранение информативности при снижении размерности данных обеспечивается регуляризацией или ограничением параметров модели, либо разработкой парсимонных (экономных) моделей. В настоящей работе для кластеризации траекторий используется метод ограничения параметров модели гауссовой смеси, уменьшающий размерность внутренних пространств кластеров и налагающий ограничения на матрицы ковариации. Этот метод относится к методам кластеризации с обучением и называется многомерным дискриминантным анализом [5, 7].

Методы кластеризации на основе модели гауссовой смеси ведут себя неоднозначно, если объем данных мал по сравнению с числом оцениваемых параметров [8]. Необходим компромисс между числом оцениваемых параметров и общностью модели. Решение этой проблемы в многомерном дискриминантном анализе состоит в повторной параметризации — рассмотрении подпространства локальных групп анализируемых объектов [5, 7].

Пусть $\mathbf{x}_j, j = \overline{1, J}$, является набором векторов размерности p , представляющих траектории $\mathbf{T}_j, j = \overline{1, J}$, в пространстве характеристик. Дискриминантный анализ является кластеризацией с обучением, целью которой является назначение вектора \mathbf{x} одному из K кластеров, известных на основе обучающей выборки. Согласно модели гауссовой смеси выборка векторов \mathbf{x} описывается распределением

$$p(\mathbf{x}, \theta) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad (1)$$

где π_k — доля k -го кластера; $\mathcal{N}(\cdot)$ — гауссово распределение вероятности со средним $\boldsymbol{\mu}_k$ и матрицей ковариации $\boldsymbol{\Sigma}_k$;

$$\begin{aligned} \mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) &= \\ &= \frac{1}{(2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right). \quad (2) \end{aligned}$$

Модель (1)—(2) соответствует методу квадратичного дискриминантного анализа, который требует оценки числа параметров $\sim p^2$, поэтому его применение затруднено в многомерном пространстве. Благодаря феномену пустого пространства [9] можно предположить, что многомерные данные локализируются в подпространствах размерности, гораздо меньшей, чем p . С учетом этого предположения в работе [10] предлагается параметризация модели гауссовой смеси, которая уменьшает число оцениваемых параметров (гауссова модель $[a_{k,j} b_k Q_k d_k]$ и ее подмодели).

Пусть \mathbf{Q}_k — ортогональная матрица со столбцами собственных векторов ковариационной матрицы $\boldsymbol{\Sigma}_k$ и Δ_k — диагональная матрица ее собственных чисел, т. е.

$$\Delta_k = \mathbf{Q}_k^T \boldsymbol{\Sigma}_k \mathbf{Q}_k. \quad (3)$$

Следовательно, матрица Δ_k — ковариационная матрица в пространстве ее собственных векторов. Предполагается, что Δ_k разделяется на два блока:

$$\Delta_k = \text{diag}(a_{k,1}, \dots, a_{k,d_k}, b_{k,1}, \dots, b_{k,p-d_k}),$$

где первые d_k ($d_k < p$) значений $a_{k,1}, \dots, a_{k,d_k}$ параметризуют дисперсию в подпространстве, содержа-

щем k -й кластер, а остальные $p - d_k$ компонент $b_{k,1}, \dots, b_{k,p-d_k}$ моделируют дисперсию шума. Модели с такой параметризацией являются парсимонными и предполагают, что каждый кластер k является изотропным. Размерность d_k рассматривается как внутренняя размерность скрытого подпространства k -й группы.

Фиксируя некоторые параметры как общие внутрикластерные или междукластерные, можно получить частные модели, соответствующие различным регуляризациям. Например, если зафиксировать первые d_k собственных значений как общие для каждого кластера, получается более ограниченная модель вида $[a_k b_k Q_k d_k]$ [10]. Применение этой модели дает хорошие результаты, т. е. предположение, что каждая матрица Δ_k содержит только два различных собственных значения a_k и b_k , является эффективным способом регуляризации оценки Δ_k . Это объясняется тем, что дисперсия при оценке a_k , которая является средним по значениям $a_{k,1}, \dots, a_{k,d_k}$, на самом деле меньше, чем дисперсия для каждой отдельной оценки $a_{k,1}, \dots, a_{k,d_k}$. Часто для сокращения внутренней размерности d_k скрытого подпространства k -й группы выбираются первые наибольшие собственные значения $a_{k,1}, \dots, a_{k,d_k}$.

Другим способом регуляризации является фиксирование параметров $b_{k,1}, \dots, b_{k,p-d_k}$, как общих между кластерами. Это приводит к моделям вида $[a_{k,j} b Q_k d_k]$ и $[a_k b Q_k d_k]$, которые предполагают, что дисперсия одинакова вне специфических подпространств кластеров. В этих моделях шум вне скрытых подпространств кластеров моделируется с помощью только одного параметра b , что оправдано, когда анализируемые данные получены одинаковым способом.

В настоящей работе при выполнении многомерного дискриминантного анализа траекторий движения самолетов испытываются различные парсимонные модели, среди которых модель $[a_k b Q_k d_k]$ оказывается наиболее эффективной. Результаты использования этой модели приводятся в разделе *Численный эксперимент*.

Численный эксперимент

В настоящей работе используются данные радара TRACON (Terminal Radar Approach Control), регистрирующего траектории полетов воздушных судов над заливом Сан-Франциско (данные находятся в открытом доступе на сайте <http://c3.nasa.gov/dash-link/resources/132/>). Подробное описание этих данных приводится в работах [2, 6]. В настоящей работе

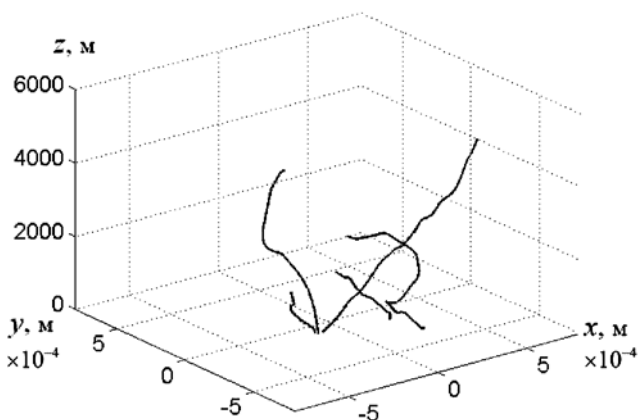


Рис. 3. Четыре посторонние траектории

анализируются 112 траекторий одинаковой длины (159 точек наблюдения) (рис. 1, см. вторую сторону обложки). Точки наблюдения самолетов фиксируются через равные промежутки времени (5 с). Каждой траектории $T_j, j = \overline{1, 112}$, соответствуют последовательности координат $X_j = \{x_j[i], i = \overline{1, 159}\}$, $Y_j = \{y_j[i], i = \overline{1, 159}\}$ и $Z_j = \{z_j[i], i = \overline{1, 159}\}$ (эти компоненты используются в многомерном дискриминантном анализе).

Для кластеризации анализируемых траекторий (рис. 2, см. вторую сторону обложки) применяется метод полиномиальных регрессий [2], и в результате, согласно сходству пространственной формы кривых (траекторий), выявляются пять кластеров траекторий, обозначенных на рис. 2 разными цветами. Помимо этого, выделяются четыре посторонние траектории (*outliers*), не принадлежащие ни к одному из выявленных кластеров (рис. 3). Разбиение всех анализируемых траекторий на кластеры вместе с набором посторонних траекторий называется совместной априорной кластеризацией.

При дискриминантном анализе траектории (см. рис. 1) разделяются на две равные выборки по 56 траекторий. Одна из выборок, демонстрирующая большую неоднородность в распределении траекторий по кластерам (см. рис. 2), используется для обучения (оценки параметров) вероятностных моделей, как это описано выше в разделе *Многомерный дискриминантный анализ*. Для каждой компоненты траекторий в трехмерном пространстве создается своя модель. Выборка, которая используется для оценки (настройки параметров) модели в многомерном дискриминантном анализе [5, 7], называется обучающей выборкой, вторая выборка — тестируемой.

На рис. 4, а–в показаны результаты покомпонентного дискриминантного анализа для компонент X , Y , и Z , соответственно (j — номер траектории в тестируемой выборке, k — номер кластера).

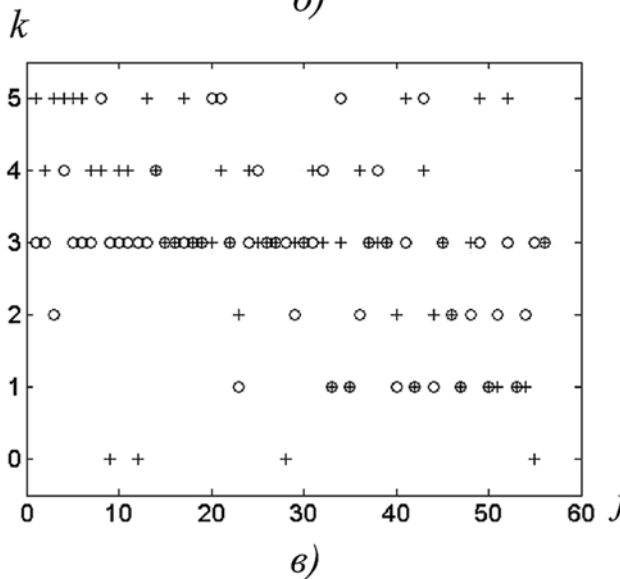
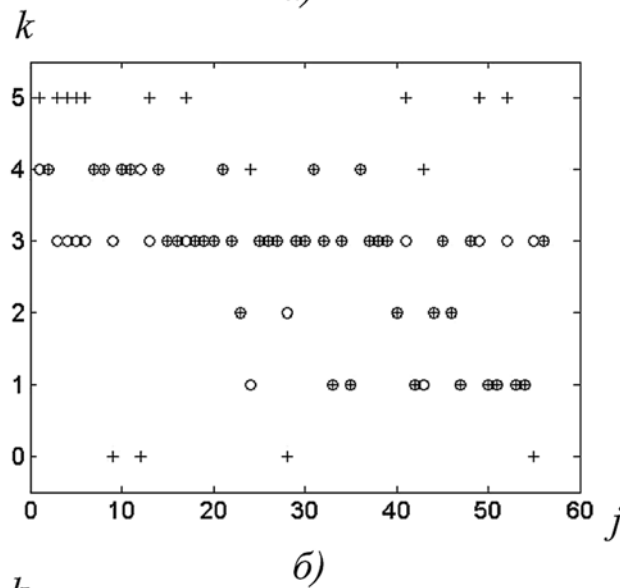
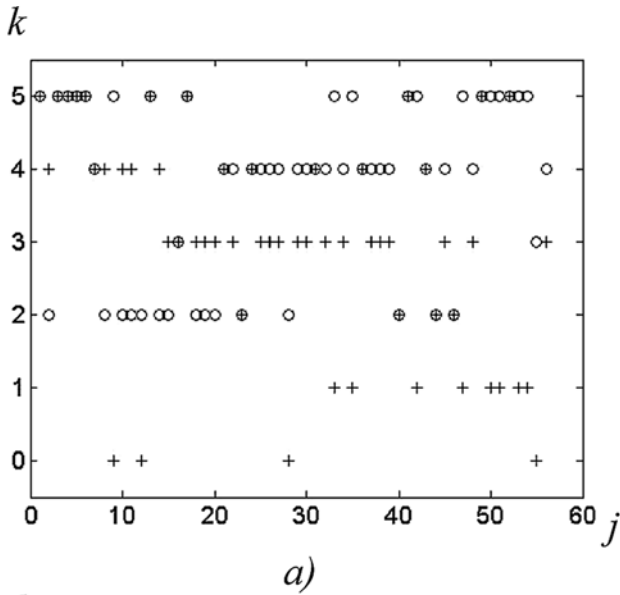


Рис. 4. Покомпонентный дискриминантный анализ результатов совместной априорной кластеризации для тестируемой выборки траекторий

Нулевому номеру кластера соответствуют посторонние траектории. Принадлежность траектории к кластеру в априорной кластеризации обозначается символом "o". Принадлежность траектории к кластеру, прогнозируемая согласно покомпонентному дискриминантному анализу, обозначается символом "+". Символ "⊕" демонстрирует совпадение априорной и прогнозируемой кластеризаций для тестируемой выборки. В результате вычисления параметров используемой модели $[a_k b Q_k d_k]$ размерности исходных данных от $p = 159$ сокращаются для пяти анализируемых кластеров следующим образом: $d_x = \{2, 1, 1, 1, 1\}$, $d_y = \{2, 1, 2, 2, 1\}$, $d_z = \{3, 1, 2, 1, 1\}$ для каждой из компонент X, Y, Z соответственно. Результаты покомпонентного многомерного дискриминантного анализа для тестируемой выборки, показанные на рис. 4, демонстрируют степень совпадения априорной кластеризации с кластеризацией, прогнозируемой согласно модели: 0,3750, 0,7143 и 0,357143 для компонент X, Y и Z соответственно. Таким образом, компонента Y выделяется как наиболее полно отражающая структуру кластеров траекторий в трехмерном пространстве. Поскольку в качестве эталона для сравнения результатов кластеризации выбирается результат кластеризации по методу полиномиальных регрессий [1, 2], основанный на естественном критерии сходства формы траекторий в пространстве, то полученные числовые оценки совпадения распределения траекторий по кластерам можно рассматривать в качестве индекса относительной информативности компонент для существующей структуры кластеров. Кроме того, отметим, что число посторонних траекторий (в данном случае — четыре), которое выявляется покомпонентным дискриминантным анализом, совпадает с числом таких траекторий, выявляемых в работах [2, 4].

Заключение

Среди различных методов кластеризации траекторий в пространстве наилучшим можно считать метод, основанный на модели полиномиальных регрессий, так как в этом случае кластеризация траекторий выполняется на основе сходства их пространственной формы. Покомпонентный дискриминантный анализ результатов кластеризации траекторий позволяет выделить одну из координатных компонент траекторий, которая является наиболее информативной для определения существующей структуры кластеров. Поскольку кластеризация траекторий в соответствии с их формой в трехмерном пространстве является некоторым эталоном, то полученные в результате покомпонентного анализа значения индекса относительной информативности компоненты позволяют сравнивать результаты различных методов кластеризации траекторий.

Список литературы

1. **Gaffney S., Smyth P.** Joint probabilistic curve clustering and alignment / Saul L., Weiss Y., Bottou L., eds. // *Proceedings of Neural Information Processing Systems (NIPS 2004)*. December 13–18, 2004. Vancouver, British Columbia, Canada. *Advances in Neural Information Processing Systems*. V. 17. Cambridge, MA: MIT Press, 2005. P. 473–480.
2. **Кухаренко Б. Г., Солнцева М. О.** Кластеризация управляемых объектов на основе сходства их многомерных траекторий // *Информационные технологии*. 2014. № 5. С. 3–7.
3. **Anjum N., Cavallaro A.** Trajectory clustering for scene context learning and outlier detection / Schonfeld D., Shan C., Tao D., Wang L., eds. // *Video Search and Mining*. Berlin. *Studies in Computational Intelligence*. V. 287. Heidelberg: Springer, 2010. P. 33–51.
4. **Кухаренко Б. Г., Солнцева М. О.** Итеративная кластеризация траекторий управляемых объектов в многомерном пространстве характеристик // *Информационные технологии*. 2014. № 8. С. 11–16.
5. **Bouveyron C., Brunet-Saumard C.** Model-based clustering of high-dimensional data: a review // *Computational Statistics & Data Analysis*. 2014. V. 71. P. 52–78.
6. **Gariel M., Srivastava A. N., Feron E.** Trajectory clustering and an application to airspace monitoring // *IEEE Transactions on Intelligent Transportation Systems*. 2011. V. 12. Is. 4. P. 1511–1524.
7. **Berge L., Bouveyron C., Girard S.** HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data // *Journal of Statistical Software*. 2012. V. 46. N 6. P. 1–29.
8. **McLachlan G., Peel D.** *Finite Mixture Models*. New York: Wiley Interscience. 2000.
9. **Scott D. W., Thompson J. R.** Probability density estimation in higher dimensions // *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*. V. 528. Amsterdam: North-Holland, 1983. P. 173–179.
10. **Bouveyron C., Girard S., Schmid C.** High-dimensional data clustering // *Computational Statistics & Data Analysis*. 2007. V. 52. N 1. P. 502–519.
11. **Jolliffe I. T.** *Principal Component Analysis*. 2nd ed. Springer Series in Statistics. New York: Springer, 2002.
12. **Guyon I., Elisseeff A.** An introduction to variable and feature selection // *Journal of Machine Learning Research*. 2003. V. 3. P. 1157–1182.

B. G. Kukharensko, Leading research scientist, Institute of Machines Science named after A. A. Blagonravov of RAS,
M. O. Solntseva, Post-graduate student, Moscow Institute of Physics and Technology (GU),
e-mail: solnceva.chalei@gmail.com

Component Wise Discriminant Analysis of Multi Dimensional Trajectory Clustering

Multi-dimensional trajectory clustering is a difficult problem due to the fact that these multi-dimensional data exist really in low-dimensional subspaces hidden in the original trajectory space. This paper uses a family of Gaussian mixture models designed for multi-dimensional data which combine the ideas of subspace clustering and parsimonious modeling. The clustering method based on the Expectation-Maximization algorithm estimates the specific subspace and the intrinsic dimension of each group. In present paper the multi dimensional discriminant analysis is applied to multi dimensional trajectory clustering based on polynomial regressions initially. An advantage of discriminant analysis is a raw data dimension reducing in subspaces representing clusters. By means of multi dimensional discriminant analysis a one trajectory projection on co-ordinate axis in source space is selected as more informative to determine cluster structure in use.

Keywords: data mining, multi dimensional trajectories, subspace clustering, Gaussian mixture models, discriminant analysis

References

1. **Gaffney S., Smyth P.** Joint probabilistic curve clustering and alignment / Saul L., Weiss Y., Bottou L., eds. *Proceedings of Neural Information Processing Systems (NIPS 2004)*. December 13–18, 2004. Vancouver, British Columbia, Canada. *Advances in Neural Information Processing Systems*. V. 17. Cambridge, MA: MIT Press, 2005. P. 473–480.
2. **Kukharensko B. G., Solntseva M. O.** Klasterizacia upravlyаемых объектов на основе shodstva ih mnogomernykh trajektoriyi. *Informacionnye tehnologii*. 2014. N 5. P. 3–7.
3. **Anjum N., Cavallaro A.** Trajectory clustering for scene context learning and outlier detection / Schonfeld D., Shan C., Tao D., Wang L., eds. *Video Search and Mining*. Berlin. *Studies in Computational Intelligence*. V. 287. Heidelberg: Springer, 2010. P. 33–51.
4. **Kukharensko B. G., Solntseva M. O.** Iterativnaja klasterizacia trajektoriyi upravlyаемых объектов v mnogomernom prostranstve harakteristik. *Informacionnye tehnologii*. 2014. N 8. P. 11–16.
5. **Bouveyron C., Brunet-Saumard C.** Model-based clustering of high-dimensional data: a review. *Computational Statistics & Data Analysis*. 2014. V. 71. P. 52–78.
6. **Gariel M., Srivastava A. N., Feron E.** Trajectory clustering and an application to airspace monitoring. *IEEE Transactions on Intelligent Transportation Systems*. 2011. V. 12. Is. 4. P. 1511–1524.
7. **Berge L., Bouveyron C., Girard S.** HDclassif: An R package for model-based clustering and discriminant analysis of high-dimensional data. *Journal of Statistical Software*. 2012. V. 46. N 6. P. 1–29.
8. **McLachlan G., Peel D.** *Finite Mixture Models*. New York: Wiley Interscience. 2000.
9. **Scott D. W., Thompson J. R.** Probability density estimation in higher dimensions. *Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface*. V. 528. Amsterdam: North-Holland, 1983. P. 173–179.
10. **Bouveyron C., Girard S., Schmid C.** High-dimensional data clustering. *Computational Statistics & Data Analysis*. 2007. V. 52. N 1. P. 502–519.
11. **Jolliffe I. T.** *Principal Component Analysis*. 2nd ed. Springer Series in Statistics. New York: Springer, 2002.
12. **Guyon I., Elisseeff A.** An introduction to variable and feature selection. *Journal of Machine Learning Research*. 2003. V. 3. P. 1157–1182.

УДК 004.4

М. А. Баранов, аспирант, **Г. В. Зеленко**, канд. техн. наук, проф.,
А. В. Рощин, канд. техн. наук, проф., e-mail: aleksey_goschin@mail.ru
Московский государственный университет приборостроения и информатики

Проблемы маршрутизации сообщений в сенсорной локальной сети

Рассмотрена проблема передачи сообщений в сенсорной локальной сети, узлы которой могут конфигурироваться одним или несколькими управляющими узлами сети. Рассмотрена графовая модель сети, позволяющая оценить одношаговую достижимость, а также общая модель прохождения сообщений между узлами сети. Для сенсорной локальной сети с одним выделенным управляющим узлом сформированы показатели комплексной оценки связности узлов сети по передаче конфигурирующих сообщений от управляющего узла к сенсорным, а также по приему измеренных данных от сенсорных узлов к управляющему. Определены соответствующие векторные показатели. Предложена процедура оценки достижимости в процессе передачи сообщений между узлами сенсорной локальной сети. Определены векторы длин путей достижимости конфигурирующих сообщений по передаче и путей достижимости сигнальных сообщений по приему. Эти значения определяют, по пути какой длины может прийти соответствующее сообщение. Рассмотрена проблема конфигурирования узлов сети для улучшения показателей достижимости.

Ключевые слова: сенсорная локальная сеть, оценка, достижимость, маршрутизация, модель

Введение

Широкое использование сенсорных локальных сетей для создания распределенных систем контроля и управления обусловлено тем, что узлы таких сетей могут конфигурироваться управляющим узлом сети [7]. Это свойство узлов сенсорной сети ставит перед проектировщиком распределенной системы управления проблему оценки достижимости при передаче сообщений в сети [4, 5]. На основании этих оценок должна вырабатываться стратегия коррекции конфигурации сети. Главное, что необходимо для успешной маршрутизации сообщений, — это наличие хотя бы одного пути, по которому это сообщение может быть передано. Одним из основных параметров, который может изменяться, является максимальная мощность передатчика узла (и связанная с ней дальность связи с соседними узлами). В статье предлагается модель оценки достижимости при передаче сообщений от одного узла сети к другому.

Модель показателей связности сенсорной локальной сети и ее узлов

Как известно, сенсорные локальные сети представляют собой ячеистую структуру, которая может быть достаточно хорошо описана графом (рис. 1).

Как и любой граф, сенсорная локальная сеть описывается парой — множеством вершин $X = [x_1, x_2, \dots, x_N]$, описывающим множество узлов сети, и присоединенной матрицей A , описывающей связи

между вершинами. Отдельно следует остановиться на нумерации узлов сенсорной локальной сети. В общем случае нумерация может быть произвольной. Однако часто в сети бывает один или несколько выделенных узлов, например управляющий узел, который собирает данные со всей сети и может координировать и конфигурировать ее узлы. В этом случае имеет смысл начинать нумерацию с одного из выделенных узлов, например с управляющего и конфигурирующего. Необходимое условие функционирования сенсорной локальной сети определяется тем, насколько достижима связь любого узла сети с любым другим.

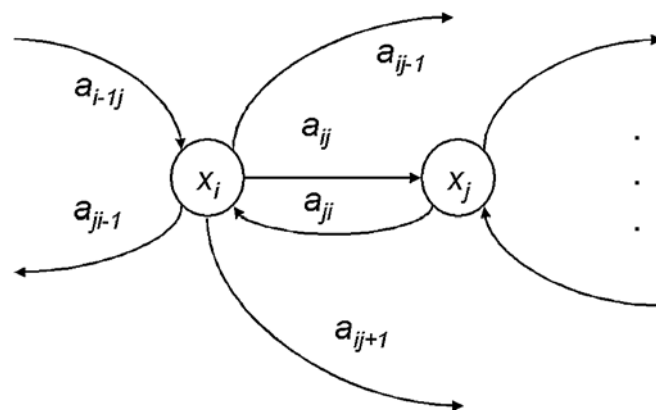


Рис. 1. Схематическое изображение фрагмента сенсорной локальной сети

Присоединенная матрица [2, 3] $A = \{a_{ij}\}_{n \times n}$ определяет связность графа модели, причем элемент a_{ij} описывает возможность передачи сообщения от элемента i элементу j . В качестве значений элементов a_{ij} могут выбираться значения, пропорциональные эффективной скорости передачи, определяемой числом потерянных пакетов, потребовавших повторной передачи. В данной статье мы будем рассматривать простейший вариант — бинарные значения: 1 — есть связь, 0 — нет связи. Здесь следует отметить, что в качестве узлов сенсорной локальной сети используются элементы, мощность передатчика которых может изменяться. То есть мощность передающей части узлов одной и той же сенсорной сети может меняться от узла к узлу. В результате может возникнуть ситуация, при которой из узла i в узел j сообщение может быть передано, а из j в i — не может. В результате присоединенная матрица графа, описывающего сенсорную локальную сеть, в общем случае будет несимметричной.

При управлении мощностью узлов сенсорной сети следует учитывать, что при уменьшении мощности передатчика уменьшаются как потребляемая мощность (и увеличивается время автономной работы элемента), так и дальность передачи сообщения. С точки зрения потребляемой мощности и времени автономной работы наилучшей присоединенной матрицей сети будет матрица, которая может быть приведена к трехдиагональной матрице (матрице, содержащей главную диагональ и две соседние). Очевидно, что сеть с такой присоединенной матрицей будет не очень надежной, так как потеря связи хотя бы одной пары узлов в цепочке приведет к потере связи между первым и последним элементами.

В общем случае матрица имеет произвольный вид [3]. Для того чтобы оценить, на какое число узлов может передать сообщение узел i , достаточно найти сумму

$$\delta_i = \sum_{s=1}^N a_{is}, \quad (1)$$

где a_i — коэффициент связности узла i по передаче; N — число узлов сенсорной локальной сети.

Аналогично, для того чтобы оценить, от какого числа узлов узел i может получить сообщения, достаточно найти сумму

$$\beta_i = \sum_{s=1}^N a_{si}, \quad (2)$$

где β_i — коэффициент связности узла i по приему.

Для комплексной оценки коэффициентов связности узлов сети можно определить векторные показатели

$$\Delta = TA; \quad (3)$$

$$B = TA^T, \quad (4)$$

где $\Delta = [\delta_1, \delta_2, \dots, \delta_N]$ — вектор оценки связности сенсорной сети по приему; $B = [\beta_1, \beta_2, \dots, \beta_N]$ — вектор оценки связности сенсорной сети по передаче; T — специально сформированная матрица размерности $n \times n$, все элементы которой — единицы. Для того чтобы оценить интегральный показатель связности сенсорной локальной сети, можно использовать показатель

$$\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_N]; \quad (5)$$

$$\lambda_i = \min\{\delta_i, \beta_i\}, i = \overline{1, N}.$$

Интегральный показатель Λ связности сенсорной локальной сети характеризует возможности как приема, так и передачи сообщений узлами сети.

Используя интегральный показатель связности сенсорной сети, можно определить слабое ее звено, из-за которого потенциально возможна потеря сообщений:

$$s = \arg(\min_i \{\lambda_i\}), \quad (6)$$

где s — номер узла сенсорной локальной сети, имеющий наименьший интегральный показатель связности либо по передаче, либо по приему, либо по тому и другому.

Узел сенсорной сети, являющийся слабым звеном, настраивается (конфигурируется) в сторону увеличения мощности передающей подсистемы. Если λ_s — показатель интегральной связности узла, являющегося слабым звеном сети, — равен нулю, это означает, что все либо передаваемые (если $\delta_s = 0$), либо получаемые (если $\beta_s = 0$) сообщения этим узлом будут потеряны.

Модель прохождения сообщений по сенсорной локальной сети

Для того чтобы оценить качество маршрутизации сообщений в сенсорной локальной сети, необходимо оценить маршруты, по которым могут проходить сообщения от каждого узла к любому другому. Если не все узлы в сенсорной сети равноправны, то имеет смысл отдельно оценить маршруты прохождения сообщений от каждого узла сети к выделенному и от выделенного узла к любому другому.

Так, например, если сенсорная локальная сеть состоит из множества узлов-датчиков, собирающих информацию для выделенного управляющего узла, который к тому же может конфигурировать остальные узлы сети, то маршруты, проложенные от каждого рядового узла сети к управляющему, позволяют оценить возможности сбора данных с датчиков, а маршруты, проложенные от управляющего узла к остальным рядовым узлам, позволят оценить возможность конфигурирования этих узлов.

Рассмотрим сенсорную локальную сеть с одним выделенным управляющим узлом, который должен собирать данные со всех остальных узлов и иметь

возможность конфигурировать эти узлы. Для определенности присвоим управляющему узлу номер 1.

В этом случае один шаг передачи конфигурирующего сообщения от первого узла будет описываться соотношением [1, 3]:

$$x_1 = Ax_0, \quad (7)$$

где $x_k = [x_k^1, x_k^2, \dots, x_k^N]^T$ — вектор состояния размерности N сенсорной локальной сети для конкретного передаваемого сообщения; x_k^i — описывает наличие передаваемого сообщения в узле i на шаге k ; $x_0 = [1, 0, \dots, 0]^T$ — начальное состояние сенсорной локальной сети при попытке передачи конфигурирующего сообщения от первого узла; x_1 — состояние сети после одного шага передачи сообщения; N — число узлов сенсорной локальной сети. Ненулевое значение i -го компонента вектора состояния означает, что сообщение присутствует в данном узле, нулевое — что его нет. При этом единичное значение говорит (с учетом определения присоединенной матрицы графа сети) о том, что сообщение передано в данный узел из одного из смежных узлов, ненулевое, отличное от единицы, значение говорит о том, что сообщение пришло более чем одним путем. При анализе одношаговой передачи сообщений значение численно совпадает с числом путей, которым оно пришло, при многошаговой передаче зависимость значения от числа путей более сложная.

В соответствии с формулой (7) состояние сенсорной сети после двух шагов передачи конкретного сообщения можно описать соотношением

$$x_2 = Ax_1 = A^2x_0, \quad (8)$$

а после выполнения p шагов —

$$x_p = A^p x_0. \quad (9)$$

Для того чтобы оценить возможность передачи конфигурирующего сообщения от управляющего (первого) узла к каждому "рядовому" узлу, необходимо выполнение следующего условия достижимости конфигурируемых сообщений:

$$\forall k \in [2 \dots N] \exists p_k \in [1 \dots N] \Rightarrow x_{p_k}^k \neq 0, x_{p_k} = A^{p_k} x_0, \quad (10)$$

где обозначения соответствуют формулам (7)–(9).

Условие (10) гарантирует достижение конфигурирующим сообщением целевого узла, причем длина пути сообщения должна быть не более N [2, 3].

Все сказанное выше относится к прохождению в сенсорной сети одного единственного сообщения.

Если в сети происходит одновременная передача многих сообщений (что соответствует реальным условиям эксплуатации сенсорных сетей), то для передаваемых сообщений справедлив принцип суперпозиции, если трафик через каждый узел не достиг ограничения.

В случае, когда на передачу сообщений влияет ограниченная пропускная способность, необходимо либо учитывать в условиях достижимости (10) имеющиеся ограничения пропускной способности каналов, либо конфигурировать узлы сенсорной локальной сети таким образом, чтобы ограничения не наступали [6].

Для анализа возможности доставки сигнальных сообщений от каждого "рядового" узла управляющему узлу следует рассмотреть следующее условие:

$$\forall k \in [2 \dots N] \exists p'_k \in [1 \dots N] \Rightarrow x_{p'_k}^1 \neq 0, x_{p'_k} = (A^T)^{p'_k} x_0; \quad (11)$$

$$x_0^k = 1, x_0^r = 0, r = [1, 2, \dots, k-1, k+1, \dots, N].$$

Условие (11) означает, что единичное сигнальное сообщение, переданное от каждого "рядового" узла сенсорной локальной сети, может быть получено управляющим (первым) узлом, причем длина пути не превышает N . Использование в условии (11) транспонированной присоединенной матрицы фиксирует тот факт, что здесь проверяется условие достижимости по приему сообщений.

Очевидно, что длина пути, по которому проходят конфигурирующие сообщения от управляющего узла к "рядовым" и сигнальных сообщений в обратном направлении, для каждого из "рядовых" узлов в общем случае различна.

Обозначим $P_{trans} = [p_2, p_3, \dots, p_N]$ вектор длин путей достижимости конфигурирующих сообщений по передаче, полученных из условия (10), а $P_{res} = [p'_2, p'_3, \dots, p'_N]$ — вектор путей достижимости сигнальных сообщений по приему, полученных из условия (11).

Очевидно, что значения компонент векторов P_{trans} и P_{res} длин путей достижимости лежат в диапазоне от 1 до N [2]. Эти значения определяют, по пути какой длины может прийти соответствующее сообщение.

Значения компонент векторов состояния сенсорной сети $x_{p_k}^k$ в условии (10) и $x_{p'_k}^1$ в условии (11) также могут иметь различное значение. В условиях достижимости (10) и (11) от них требуется лишь неравенство нулю, что гарантирует принципиальную достижимость для соответствующих сообщений. Если соответствующая компонента в указанных векторах состояния равна 1, это значит, что сообщение может пройти по единственному пути. Если значение компоненты более 1, это означает, что для прохождения соответствующего сообщения имеется более одного пути. Для того чтобы оценить число возможных путей, придется восстановить маршруты прохождения этих сообщений.

Для определения возможных маршрутов прохождения сообщений можно применить следующую процедуру.

Для простоты рассмотрим процедуру для определения маршрутов конфигурирующих сообщений согласно условию (10). Понятно, что процедура должна выполняться для каждого "рядового" узла сети.

1. Определяется p_k , при котором впервые появляется ненулевой элемент $x_{p_k}^k$, затем от конца строится возможный маршрут прохождения сообщения.

2. Ищутся возможные предшествующие узлы сети — это все узлы, для которых в k -й строке транспонированной присоединенной матрицы A^T присутствуют ненулевые элементы. Каждый из этих узлов $\{\sigma_1, \sigma_2, \dots, \sigma_p\}$ является кандидатом на элемент возможного маршрута доставки сообщения. В данном случае считается, что таких возможных узлов p штук. Очевидно, что $p \leq N$. Останется ли найденный кандидат в возможном маршруте или нет, зависит от того, дойдет ли проверка до первого (управляющего) узла.

3. Для каждого из найденных в предыдущем пункте узлов таким же методом ищутся все возможные предшествующие узлы сенсорной сети, которые также становятся претендентами для возможного маршрута доставки конфигурирующего сообщения. Критерий оставления кандидатов возможного маршрута тот же.

4. Действия, указанные в п. 3, повторяются в общей сложности p_k раз. В результате в получившемся дереве маршрутов оставляются только те ветви, которые приводят к первому (управляющему) узлу. Совокупность путей, составленных из оставленных ветвей построенного дерева, и представляют собой множество маршрутов, по которым конфигурирующее сообщение может достигнуть анализируемого узла сети.

5. После проведения указанной процедуры для каждого из "рядовых" узлов сети получаем множество возможных маршрутов достижения цели конфигурирующими сообщениями.

Множество маршрутов достижения выделенного управляющего узла сигнальными сообщениями от "рядовых" узлов ищется аналогичным способом.

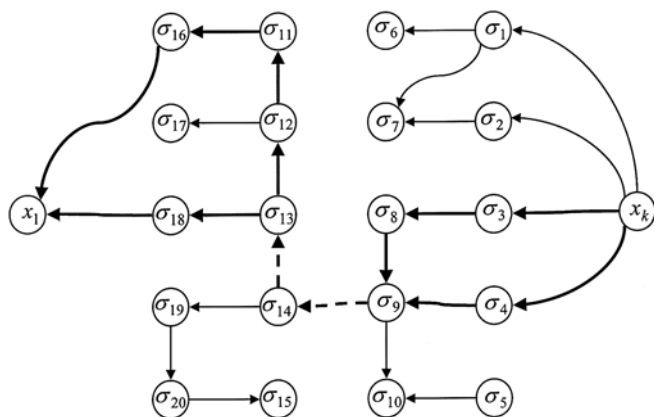


Рис. 2. Пример дерева достижимости управляющего узла сигнальными сообщениями

Пример дерева, построенного для узла k , показан на рис. 2.

На рис. 2 показано найденное дерево возможных маршрутов передачи сигнальных сообщений от узла k . На рисунке жирными стрелками выделены дуги графа, которые определяют возможные маршруты, доходящие до первого (управляющего) узла. Это маршруты:

$$x_k - \sigma_3 - \sigma_8 - \sigma_9 - \sigma_{14} - \sigma_{13} - \sigma_{18} - x_1,$$

$$x_k - \sigma_4 - \sigma_9 - \sigma_{14} - \sigma_{13} - \sigma_{18} - x_1,$$

$$x_k - \sigma_3 - \sigma_8 - \sigma_9 - \sigma_{14} - \sigma_{13} - \sigma_{12} - \sigma_{11} - \sigma_{16} - x_1,$$

$$x_k - \sigma_4 - \sigma_9 - \sigma_{14} - \sigma_{13} - \sigma_{12} - \sigma_{11} - \sigma_{16} - x_1.$$

Штриховыми жирными линиями показаны дуги графа, являющиеся общими для всех возможных маршрутов. Эти дуги $(\sigma_9 - \sigma_{14} - \sigma_{13})$ являются критическими секциями найденных маршрутов. Это значит, что если хотя бы одна из этих дуг $(\sigma_9 - \sigma_{14})$ или $(\sigma_{14} - \sigma_{13})$ исчезнет, пути доставки сигнальных сообщений из узла k в первый узел перестанут существовать. Если в найденных маршрутах обнаруживаются такие критические секции, имеет смысл таким образом изменить конфигурацию узлов, входящих в эти секции, чтобы появились дублирующие сегменты маршрутов. Для этого можно, например, увеличить мощность передатчиков этих узлов.

Заключение

В статье построены оценки достижимости для сообщений, передаваемых в сенсорной локальной сети в зависимости от конфигурации сети. Эти оценки могут использоваться для выработки стратегии коррекции конфигурации сети. Предложена также модель прохождения сообщений от одного узла сети к другому, а также предложена процедура проведения коррекции конфигурации сети для исключения "критических" секций сети и уменьшения вероятности потери пакетов при передаче.

Список литературы

1. Саати Т. Л. Математические модели конфликтных ситуаций. М.: Советское Радио, 1977. 304 с.
2. Оре О. Теория графов. М.: Наука, 1980. 336 с.
3. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978.
4. Зеленко Г. В., Рошин А. В., Третьяков С. В. Проектирование встраиваемых управляющих систем методом ситуационного анализа // Полет. 2008. № 10. С. 56–60.
5. Смирнов И. В., Зеленко Г. В., Рошин А. В. Особенности разработки комплекса микроконтроллерных устройств // Промышленные АСУ и микроконтроллеры. 2012. № 1. С. 48–50.
6. Смирнов И. В. Методика комплексной разработки микроконтроллерных устройств // Промышленные АСУ и микроконтроллеры. 2012. № 11. С. 54–57.
7. Семенов А. Беспроводные сенсорные сети — новые модели использования. URL: <http://www.bytemag.ru/articles/detail.php?ID=8780>

Message Routing Problems in Sensory LAN

This paper describes the reachability problem in the transmission of messages between sensor network nodes, which can be configured by the operating node of a network. The graph model of the network, allowing to estimate single-step reachability, and also the general model of passing messages between network nodes is considered. For a sensor network with one allocated operating node indicators of a complex assessment of connectivity in transfer of configuring messages from the operating node to sensor node, and also in reception of the measured data from sensor nodes to the operating one are created. The corresponding vector indicators are defined. Procedure of an assessment of reachability in the course of transmission of messages between nodes of a sensor local network is offered. Vectors of lengths of transfer reachability paths for configuring messages and reception reachability paths for alarm messages are determined. These values define the length of the path for the corresponding message. The problem of a configuration of network nodes for improvement of reachability is considered.

Keywords: sensory LAN, assessment, reachability, routing model

References

1. Saati T. L. *Mathematicheskie modeli konfliktnykh situacij*. M.: Sovetskoe Radio, 1977. 303 p.
2. Ore O. *Teoriya grafov*. M.: Nauka, 1980. 336 p.
3. Kristofides N. *Teoriya grafov. Algoritmicheskij podhod*. M.: Mir, 1978.
4. Zelenko G. V., Roshchin A. V., Tret'jakov S. V. *Proektirovanie vstraivaemykh upravljajushchih sistem metodom situacionnogo analiza*. *Polet*. 2008. N. 10. P. 56–60.
5. Smirnov I. V., Zelenko G. V., Roshchin A. V. *Osobennosti razrabotki kompleksa mikrokontrollernykh ustrojstv. Promyshlennye ASU i mikrokontrollery*. 2012. N. 1. P. 48–50.
6. Smirnov I. V. *Metodika kompleksnoj razrabotki mikrokontrollernykh ustrojstv. Promyshlennye ASU i mikrokontrollery*. 2012. N. 11. P. 54–57.
7. Semenov A. *Besprovodnye sensornye seti — novye modeli ispol'zovaniya*. URL: <http://www.bytemag.ru/articles/detail.php?ID=8780>

УДК 004.624

С. В. Поршнеv, д-р техн. наук, проф., Д. А. Божалкин, аспирант, e-mail: bozhalkin@mail.ru
ФГАОУ ВПО "Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина", г. Екатеринбург

Технология семантического анализа дампа трафика информационных потоков в компьютерных сетях

Рассматриваются различные подходы к решению задачи семантического анализа дампов трафика информационных потоков в компьютерных сетях. Описаны особенности дампов трафика. Изучены возможности математических инструментов для работы с данными файлов дампа. Получен метод анализа дампов трафика, имеющий неограниченные возможности по извлечению параметров протоколов любых уровней, и проведена проверка его работоспособности.

Ключевые слова: трафик, семантический анализ, дампы, компьютерные сети, сетевые протоколы, сетевой пакет, математические инструменты, извлечение данных, обработка данных

Введение

Теория массового обслуживания являлась основной для систем распределения информации вплоть до 1993 г., когда после исследований, проведенных W. Leland, M. Taqqu, W. Willinger и D. Wilson [1], становится ясным, что потоки в современных сетях имеют иную структуру, чем принято в классической теории телетрафика. Как следствие, расчет параметров системы распределения информации, пред-

назначенной для обработки сетевого трафика, по классическим формулам приводит к получению недостоверных результатов и, соответственно, неправильным рекомендациям по выбору технических характеристик сетевого оборудования.

При этом необходимо отметить, что несмотря на значительное число ученых, занимавшихся изучением данной проблемы, и большое количество опубликованных работ, ее окончательного решения

пока не получено. В том числе отсутствуют: общепризнанная модель Интернет-трафика; строгие теоретические обоснования способов точного расчета технических характеристик современных систем передачи информации в Интернет-каналах и показателей качества обслуживания пользователей данных систем; устоявшиеся методики получения и анализа реального Интернет-трафика. В этой связи задачи дальнейшего изучения свойств трафика, особенно в магистральных высокоскоростных Интернет-каналах, и совершенствования на их основе математических моделей Интернет-трафика являются актуальными.

Отсутствие готового программного инструмента, позволяющего проводить комплексные исследования дампов трафика, включая и математическую обработку данных, с неизбежностью создавало значительные трудности в ходе проведения подобных исследований. Для решения данной проблемы в статье предложена технология работы с большими объемами экспериментально получаемой в сети Интернет информации (трафик реальных магистральных каналов), передаваемой за определенный промежуток времени, — дампами трафика, позволяющая работать, используя единственный программный инструмент — пакет MATLAB.

1. Анализ особенностей объекта исследования

В качестве объекта исследований были выбраны ежедневные пятнадцатиминутные дампы трафика магистрального канала между США и Японией, находящиеся в общедоступном хранилище — MAWI (Measurement and Analysis on the WIDE Internet) [2]. Здесь находится как актуальная информация о текущем состоянии Интернет-канала, так и архивные данные, начиная с 1999 г. Для сохранения конфиденциальности информации в дампе обезличены сведения о конкретных пользователях. Кроме того, содержание пакетов заменено на сгенерированные случайным образом данные, которые, однако, хранят в себе всю необходимую для исследований техническую информацию (номера портов, протоколы, размер пакета и т. д.).

Дамп сетевого трафика можно рассматривать как "снимок данных", прошедших через сеть или сетевое устройство за определенный момент времени. Сегодня стандартом записи потока сетевого трафика, де-факто, является формат .pcap (Packet Capture). С данным форматом работает большинство сетевых анализаторов, в частности, такие известные программные инструменты как tcpdump [3] и Wireshark [4], которые предназначены для работы с трафиком (запись, чтение дампов и др.), передаваемым в компьютерных сетях с различной скоростью передачи данных. Также для работы с pcap-файлами используются библиотеки libpcap и winpcap [5], работающие под управлением ОС Unix и ОС Windows, соответственно, которые совершенствуются одно-

временно с модернизацией сетевого оборудования (например, в их последней версии 2.4 была добавлена поддержка времени с точностью до 1 нс).

Pcap-файл имеет заголовок, содержащий в себе глобальную информацию, за которым записывается информация, содержащаяся в каждом захваченном пакете, или ноль, если в данный временной интервал не было получено ни одного пакета. Формат записи захваченных пакетов зависит от протокола и используемой технологии передачи данных (Ethernet, ATM и т. д.). Структура pcap-файла представлена на рис. 1.

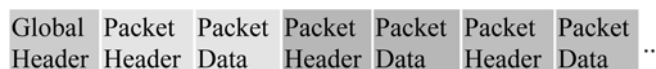


Рис. 1. Структура pcap-файла

Глобальный заголовок, за которым следует заголовки первого пакета, содержит в себе следующую информацию:

```
typedef struct pcap_hdr_s {
    guint32 magic_number; /*магический номер*/
    guint16 version_major; /*номер основной версии*/
    guint16 version_minor; /*номер промежуточной
                             версии*/
    guint32 thiszone; /*часовой пояс с поправкой на GMT (среднее время по Гринвичу)*/
    guint32 sigfigs; /*точность временных меток*/
    quint32 snaplen; /*максимальная длина пойманных пакетов, в октетах*/
    guint32 network; /*тип передачи данных*/
} pcap_hdr_t;
```

Здесь:

— magic_number (магическое число) — число, используемое для определения формата файла и порядка следования байтов. (Например, если программа сетевого анализатора, захватывающая трафик в файл дампа, имеет в данной переменной 0xa1b2c3d4 в оригинальном формате и соответствующем порядке байтов, то программа сетевого анализатора, читающая этот файл дампа, будет считывать последовательность байтов, аналогичную записанной последовательности — 0xa1b2c3d4. Если последовательность магического числа обратная — 0xd4c3b2a1, приложение, читающее файл дампа, знает, что считывается измененное значение магического числа 0xd4c3b2a1, а потому необходимо поменять местами соответствующие поля);

— version_major, version_minor (основная, промежуточная версии) — номера соответствующих версий формата текущего файла;

— thiszone (часовой пояс) — время коррекции в секундах между средним временем по Гринвичу (GMT) или всемирным координированным временем (UTC) и временем текущего часового пояса для временных меток заголовка очередного пакета. (Например, если временные метки зафиксированы по GMT или UTC, то часовой пояс будет равняться нулю. Если временные метки зафиксированы для центрально-европейского времени (Амстердам, Берлин и др.), которое составляет GMT + 1:00, часовой пояс должен иметь значение -3600. На практике всегда используются отметки времени в формате GMT, и часовой пояс всегда равен 0);

— sigfigs (точность временных меток) — параметр, задающий точность, с которой будут определяться временные метки.

— snaplen (максимальная длина пойманных пакетов) — величина, определяющая, сколько октетов каждого пакета необходимо захватывать (обычно используется значение, гарантирующее захват всех октетов для любого пакета, равное 65 535);

— network (тип способа передачи данных) — параметр, выбираемый в соответствии с типом заголовка канального уровня и типом заголовка для каждого пакета (например, 802.11, 802.11 с различной уточняющей информацией, PPP, Token Ring, FDDI и т. д.).

Пример окна программы Wireshark, в котором представлено содержимое дампа MAWI от 07.02.2014

[6], приведен на рис. 2. Из рис. 2 видно, что файл дампа сетевого трафика содержит в себе исчерпывающую информацию о данных, передаваемых в сети. Следовательно, задача дальнейшего анализа дампа Интернет-трафика относится к задачам полного синтаксического анализа (парсингу) полученных файлов.

Необходимо отметить, что сегодня существует большое количество различных сетевых анализаторов Интернет-трафика, обеспечивающих создание и просмотр дампов сетевого трафика, однако данные инструменты не позволяют проводить полноценный с математической точки зрения анализ свойств трафика. Кроме того, у обсуждаемых программных средств отсутствует возможность непосредственной передачи информации, извлеченной из дампа, в какой-либо математический пакет для ее последующей обработки.

Основная причина, обусловившая описанную выше ситуацию, состоит в том, что размеры дампов, содержащих исходную информацию о трафике, оказываются столь большими (несколько сотен мегабайт и более), что стандартные средства для работы с данными файлами и анализа их содержимого оказываются неработоспособными. В этой ситуации понятно, что необходима разработка соответствующей технологии, обеспечивающей возможность полноценного использования дампа.

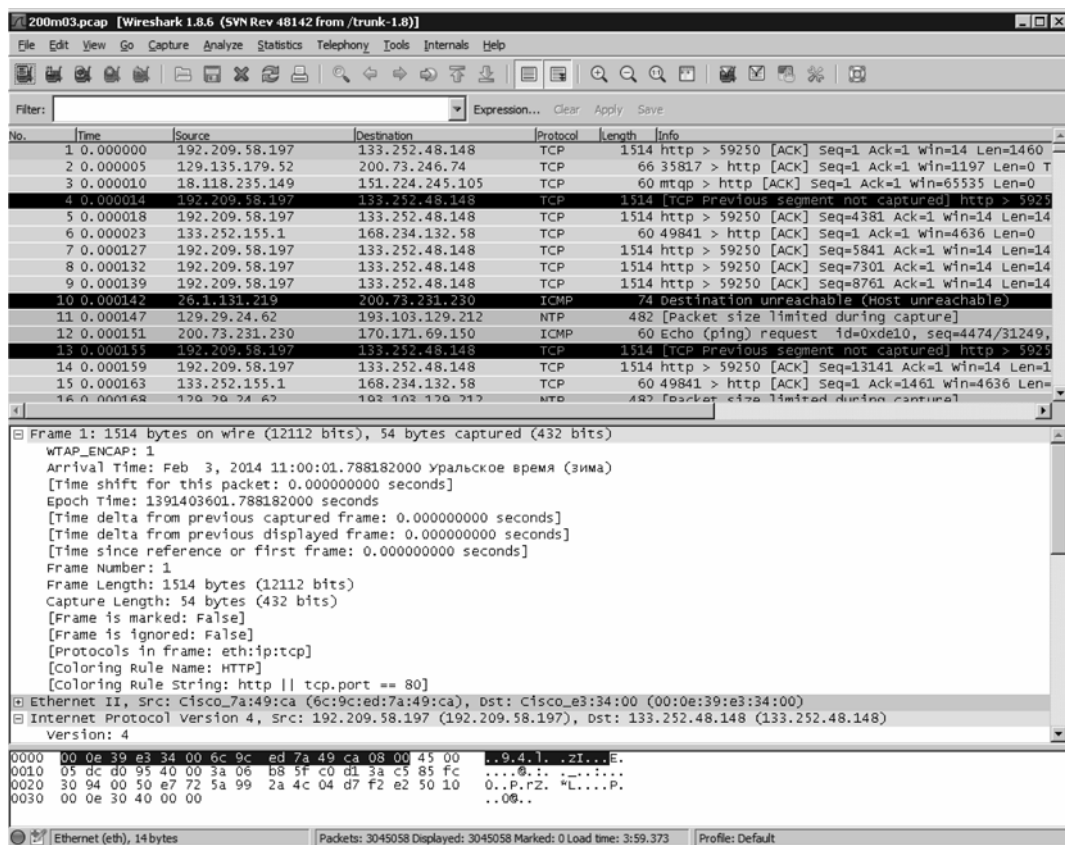


Рис. 2. Состояние рабочего окна программы Wireshark в ходе анализа дампа сетевого трафика от 07.02.2014 [6]

2. Технология работы с дампом

2.1. Обоснование выбора математического инструмента анализа дампа трафика

Разработку технологии работы с дампом целесообразно начать с выбора программного инструмента, в котором реализованы математические методы обработки данных. Перечень изученных программных продуктов, которые потенциально можно использовать для обработки дампа трафика, представлен в табл. 1.

Из табл. 1 видно, что рассмотренные программные продукты в целом являются достаточно сходными по реализованному в них функционалу. При этом пакеты Scilab [7], Octave [8], Sage [9] являются свободно распространяемыми программными продуктами. Однако у данных пакетов отсутствует техническая поддержка, кроме того, существуют проблемы с кодировкой, у них меньшее количество доступных расширений, дополнений и реализованных в них математических функций [11, 12]. Пакет MATLAB [10] не имеет недостатков реализации, перечисленных у конкурентов. MATLAB является платным коммерческим продуктом, поэтому его

основной недостаток в сравнении с конкурентами — это цена. Данный недостаток MATLAB является несущественным, так как цена для студентов/аспирантов составляет 45 долларов (~1500 руб.). Также MATLAB является инструментом, который сегодня активно используется как членами научного сообщества, так и профессионалами в различных сферах человеческой деятельности. В связи с этим в качестве базового программного средства, используемого для математического анализа дампа, был выбран MATLAB.

2.2. Обоснование выбора подходов и инструментов лингвистического анализа дампа трафика

Для выбора средства лингвистического анализа (парсинга) дампа было приведено сравнение возможных подходов к его реализации (табл. 2).

Классическим вариантом решения задачи обработки данных дампа является схема: дампы → парсер → файл → программа обработки. Очевидно, что данный подход предусматривает использование промежуточных операций: запись данных в файл и чтение соответствующих данных из файла, что делает его неэффективным. Можно существенно со-

Таблица 1

Сравнительная таблица средств математической обработки данных

Критерии сравнения	Наименования программных пакетов			
	Scilab	Octave	Sage	MATLAB
Построение графиков	+	Использует дополнительные программы: <i>Gnuplot</i> и <i>Grace</i>	+	+
Возможность имитационного моделирования	+	—	—	+
Функционал	Большое количество готовых функций	Большое количество готовых функций	Ограниченное количество готовых функций	Большое количество готовых функций и дополнений
Использование вычислительных ресурсов	Способен решить методом Гаусса СЛАУ 256x256	Способен решить методом Гаусса СЛАУ 2048x2048	Способен решить методом Гаусса СЛАУ 1024x1024	Способен решить методом Гаусса СЛАУ 2048x2048
Возможность компиляции других языков	C, C++, Fortran	C, C++, Fortran	Phyton	C, C++, Visual C, Fortran
Удобство установки и настройки	Не требует дополнительных настроек после установки	Необходима дополнительная настройка для запуска графического интерфейса	Необходима дополнительная настройка шрифтов для корректного отображения в графическом окне	Не требует дополнительных настроек после установки
Совместимость с операционными системами	Linux, MacOS X, Windows	Linux, UNIX, Cygwin, Windows, Android	Linux, Unix, Windows	Microsoft Windows, Mac OS, Linux, Unix
Разработчик	Scilab Enterprises	John W. Eaton	Уильям Стей	The MathWorks
Техническая поддержка	Нерегулярные обновления. Получение информации на форумах и в сообществах	Нерегулярные обновления. Получение информации на форумах и в сообществах	Нерегулярные обновления. Получение информации на форумах и в сообществах	Регулярные обновления. Получение информации онлайн по телефону, а также на форумах и в сообществах
Стоимость	Бесплатно. Лицензия GNU GPL	Бесплатно. Лицензия CeCILL (совместима с GNU GPL v.2)	Бесплатно. Лицензия GNU GPL	1500 руб. для студентов. Лицензия проприетарная

Сравнительная таблица средств парсинга дампа

Критерии сравнения	Подходы к реализации лингвистического анализа			
	Написать парсер самостоятельно	Импортировать дампы вручную	Traceplay	Sharktools
Открытый исходный код	+	—	—	+
Возможности распознавания	Только необходимые заранее известные параметры для определенных протоколов	Только необходимые заранее известные параметры для определенных протоколов	Ограниченное число параметров	Неограниченные возможности по извлечению параметров протоколов любых уровней
Временные затраты на реализацию	Целесообразно только при необходимости извлечь определенные заранее известные параметры	Готовое решение (использование стандартной функции импорта MATLAB)	Компиляция в виде <i>mex</i> -функции средствами MATLAB	Компиляция в виде <i>mex</i> -функции средствами MATLAB
Количество операций, которые необходимо совершить вручную для переноса данных в среду MATLAB	1. Подготовка дампа для импорта 2. Импорт дампа	1. Подготовка дампа для импорта (перевод в текстовый формат) 2. Импорт дампа 3. Правка некорректно импортированных данных	1. Сформировать запрос из командной строки MATLAB	1. Сформировать запрос из командной строки MATLAB
Быстродействие	Средняя скорость: данные передаются в среду MATLAB после предварительной подготовки	Очень медленно: большой процент некорректно импортированных данных, данные попадают в среду MATLAB после предварительной подготовки	Самый быстрый способ: данные попадают в среду MATLAB напрямую	Самый быстрый способ: данные попадают в среду MATLAB напрямую
Стоимость	Бесплатно	Бесплатно	Бесплатно в образовательных целях. Лицензия проприетарная	Бесплатно. Лицензия GNU GPL

кратить число промежуточных операций, если реализовать прямую передачу данных из парсера в программу обработки, т. е. использовать схему: дампы → парсер → программа обработки.

Для реализации предложенного подхода были рассмотрены возможности библиотеки Traceplay [13] и программы Sharktools [14]. К преимуществам использования библиотеки Traceplay следует отнести возможность загрузки дампа трафика непосредственно в среду MATLAB, к недостаткам — отсутствие открытого исходного кода и ограниченное число

параметров, которые можно извлечь с помощью данной библиотеки. К несомненным достоинствам программы Sharktools [14], с точки зрения решаемой задачи, следует отнести возможности использования технологии фильтров (DisplayFilters) пакета синтаксического анализа дампа трафика Wireshark [15], а также создания *mex*-функций MATLAB. *Mex*-функция MATLAB, которая в соответствии с выбранным фильтром напрямую обращается к файлу дампа трафика, проводит его синтаксический анализ средствами пакета Wireshark и передает его результаты в рабочее пространство пакета MATLAB. Структурная схема предложенной технологии представлена на рис. 3.

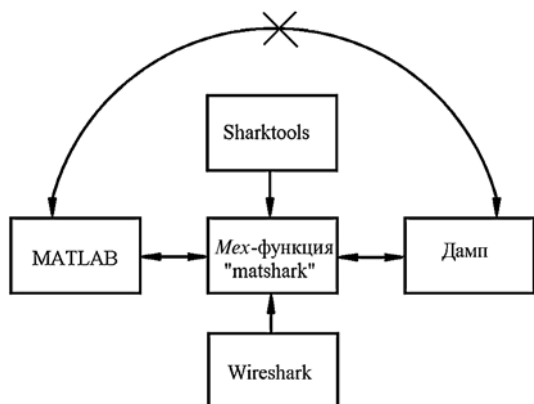


Рис. 3. Организация импорта данных из дампа в среду обработки MATLAB

2.3. Обоснование выбора операционной системы

Важность выбора операционной системы обусловлена тем, что работа с дампом сетевого трафика является достаточно ресурсоемкой задачей, поэтому здесь целесообразно использовать такую ОС, которая потребляет на собственные нужды минимум вычислительных ресурсов.

Проведенный анализ известных ОС показал, что ОС семейства Windows (7, 8, 2008 server, 2012 server) [16] предъявляют следующие минимальные требования к вычислительным ресурсам:

— процессор с тактовой частотой 1 ГГц или выше;

Сравнительная таблица Unix-подобных операционных систем

Критерии сравнения	Операционные системы			
	Ubuntu Server	FreeBSD	OS X	Red hat
Техническая поддержка (регулярные обновления)	+	+	+	+
Поддерживаемое оборудование	Оборудование с x32/x64 архитектурой	Оборудование с x32/x64 архитектурой	Работает только на оборудовании разработчика ОС	Оборудование с x32/x64 архитектурой
Поддержка программных пакетов, выбранных нами для исследования	+	—	+	+
Стоимость	Бесплатно. Лицензия GNU GPL	Бесплатно. Лицензия BSD	Бесплатно (текущая версия — Maveric). Лицензия проприетарная	От 1500 руб. Лицензия проприетарная

— 1 Гбайт (для 32-разрядной системы) или 2 Гбайт (для 64-разрядной системы) оперативной памяти (ОЗУ);

— 16 Гбайт (для 32-разрядной системы) или 20 Гбайт (для 64-разрядной системы) пространства на жестком диске;

— Unix-подобные ОС (Ubuntu, BSD, OS X, Red hat и т. д.) [17];

— процессор с тактовой частотой 0,8 ГГц или выше;

— 512 Мбайт оперативной памяти (ОЗУ);

— 5 Гбайт пространства на жестком диске.

В связи с этим был сделан выбор в пользу Unix-подобных ОС. В табл. 3 приведены основные характеристики наиболее актуальных из них.

Из табл. 3 видно, что наиболее подходящей ОС является Ubuntu Server [17, 19], которая используется далее для проверки работоспособности предложенной технологии.

3. Результаты экспериментальной проверки работоспособности технологии работы с дампом

Для проверки работоспособности предложенной технологии работы с дампом трафика на основе результатов анализа инструментов математической обработки, методов лингвистического анализа, сетевых анализаторов и операционных систем был выбран следующий перечень программ и аппаратных компонентов.

1. Операционная система — Ubuntu Server 12.04.4 LTS.

2. Инструмент математического анализа — MATLAB 2013b.

3. Сетевой анализатор — Wireshark 1.10.05.

4. Программа для создания *tex*-функции — Sharktools 0.1.6.

5. ЭВМ следующей конфигурации:

5.1. Процессор Intel Xeon X5670 2.93 ГГц (2 ядра);

5.2. 16 Гбайт оперативной памяти;

5.3. Жесткий диск объемом 500 Гбайт.

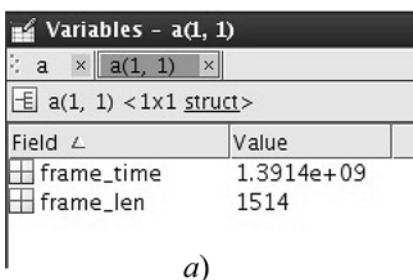
В качестве исходной информации использовались файлы с дампом трафика, размещенные на сайте [2], из которых с помощью *tex*-функции пакета MATLAB *matshark*, разработанной авторами, выбиралась необходимая для целей дальнейшего анализа информация. Данная функция имеет следующий синтаксис:

a = matshark('Название файла', {'Извлекаемый параметр1', 'Извлекаемый параметр2', ...}).

При вызове названия извлекаемых параметров должны совпадать с форматом соответствующего фильтра программы Wireshark. Далее извлеченная информация передается функцией *matshark* непосредственно в рабочее пространство MATLAB в виде упорядоченной структуры (массива двумерных ячеек, первое поле которых содержит название параметра, извлеченного из дампа, вторая ячейка — значение данного параметра). Состояние рабочего пространства MATLAB после выполнения команды

a = matshark('MAW102072014.pear', {'frame.time', 'frame.len'})

представлено на рис. 4. Здесь в качестве объекта исследования был использован дамп трафика магистрального Интернет-канала от 07.02.2014 [6] из



а)

a (1,1); a (1,2) ... a (1,n)
↓
(имя параметра (1), значение параметра (1))
(имя параметра (2), значение параметра (1))
...
(имя параметра (n), значение параметра (n))

б)

Рис. 4. Данные из файла дампа, переданные в рабочее пространство MATLAB функцией *matshark*:

a — состояние рабочего пространства MATLAB после окончания работы функции *matshark* в MATLAB; *b* — структура массива двумерных ячеек

хранилища MAWI. MA71. Размер файла, содержащего 15-минутный снимок дампа трафика магистрального канала, составляет 4,72 Гбайт.

Из рис. 4 видно, что каждая ячейка структуры содержит время прибытия соответствующего пакета (переменная `frame_time`) и его размер (переменная `frame_len`). Время в файле формата `.pcap` указано в секундах, прошедших с полуночи по UTC 01.01.1970 до времени поступления пакета.

Наличие данных о времени прибытия пакета и его размере позволяет вычислить объем переданной за каждую секунду информации (рис. 5). (Отметим, что анализ дампа необходимо проводить, начиная со второй секунды и заканчивая предпоследней, так как информация о первой и последней секунде может оказаться неполной и, следовательно, невозможно достоверно определить объем передаваемых в этот момент данных.)

Типичные результаты оценки временных характеристик, полученные в ходе тестирования предложенной технологии, представлены в табл. 4.

Так как зависимость времени выполнения запроса на извлечение требуемых данных из дампа от числа запрашиваемых параметров и размера самого дампа является линейной (это следует из самого характера обработки), то исходя из данных табл. 4, можно оценить приблизительно время обработки дампов различных размеров.

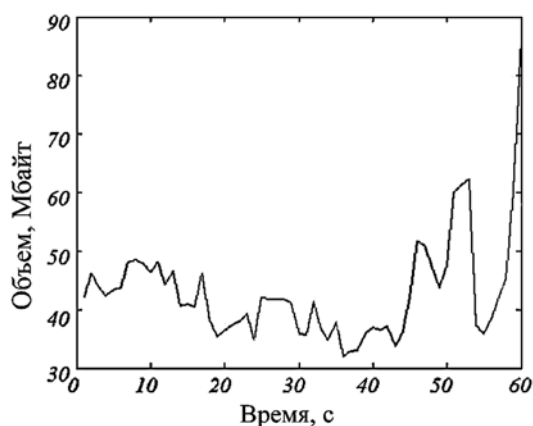


Рис. 5. Объем информации, переданной за каждую секунду на интервале 60 с

Таблица 4

Время обработки дампа в зависимости от его размера и числа извлекаемых параметров

Размер дампа, Мбайт	Число извлекаемых параметров	Время обработки, мин
200	2	2...5
400	2	5...10
600	2	10...15
$1 \cdot 10^3$	2	25...30
$1 \cdot 10^3$	4	55...65
$2 \cdot 10^3$	2	55...65

Заключение

Предложена технология анализа дампов трафика, регистрируемого в компьютерных сетях, которая путем интеграции функции парсинга в программный пакет MATLAB позволяет извлекать данные из дампа и проводить его исследования с помощью единственного программного средства.

Работоспособность предложенной технологии подтверждается результатами ее применения для анализа дампов реального трафика, передаваемого в высокоскоростном магистральном Интернет-канале. Данная технология будет использована для последующего анализа и математического моделирования потоков сетевого трафика высокоскоростных магистральных Интернет-каналов. Изложение их результатов является предметом последующих публикаций.

Список литературы

1. Leland W. E., Taquq M. S., Willinger W., Wilson D. V. On the Self-Similar Nature of Ethernet Traffic // Ref. Libr. 1994. Vol. 2, N 1. P. 1—15.
2. MAWI Working Group Traffic Archive. URL: <http://mawi.wide.ad.jp/mawi> (дата обращения: 13.03.2014).
3. Tcdump/Libcap public repository. 2010—2014. URL: <http://www.tcpdump.org> (дата обращения: 13.03.2014).
4. Wireshark. URL: www.wireshark.org (дата обращения: 13.03.2014).
5. Network media specific capturing // Wireshark. URL: <http://wiki.wireshark.org/CaptureSetup/NetworkMedia> (дата обращения: 13.03.2014).
6. Traffic Trace Page. URL: <http://mawi.nezu.wide.ad.jp/mawi/samplepoint-F/2014/201401071400.dump.gz> (дата обращения: 13.03.2014).
7. Scilab — Open source software for numerical computation. URL: www.scilab.org (дата обращения: 13.03.2014).
8. GNU Octave // Операционная система GNU. 1998—2013. URL: <https://www.gnu.org/software/octave> (дата обращения: 13.03.2014).
9. Sage: Open Source Mathematics Software [Электронный ресурс]. URL: www.sagemath.org (дата обращения: 13.03.2014).
10. Mathworks — MATLAB and Simulink for Technical Computing. 1994—2014. URL: www.mathworks.com (дата обращения: 13.03.2014).
11. Sharma N., Gobbert M. K. A comparative evaluation of MATLAB, octave, freemat, and scilab for research and teaching. University of Maryland. Baltimore County. 2010.
12. Sharma N. A comparative study of several numerical computational packages. M. S. thesis. University of Maryland. Baltimore County, 2010.
13. Tracesplay. 1999—2008. URL: tracesplay.sourceforge.net (дата обращения: 13.03.2014).
14. Armenb/sharktools // GitHub — Build software better, together. URL: <https://github.com/armenb/sharktools> (дата обращения: 13.03.2014).
15. DisplayFilters // Wireshark. URL: <http://wiki.wireshark.org/DisplayFilters> (дата обращения: 10.04.2014).
16. Системные требования ОС Windows 7 — Microsoft Windows // Microsoft Windows. URL: <http://windows.microsoft.com/ru-ru/windows7/products/system-requirements> (дата обращения: 13.03.2014).
17. Установка Ubuntu Linux: инструкция для пользователя Windows // Русскоязычная документация по Ubuntu. URL: http://help.ubuntu.ru/wiki/ubuntu_install (дата обращения: 13.03.2014).
18. Web-Cite query result. // WebCite. URL: <http://www.webcitation.org/64vYth5tU> (дата обращения: 13.03.2014).
19. The world's most popular free OS | Ubuntu. URL: www.ubuntu.com (дата обращения: 13.03.2014).

Semantic Analysis Technology of Dataflow in Computer Networks Traffic Dump

Different approaches to solving the problem of semantic analysis of dataflow in computer networks traffic dumps are considered. The features of traffic dumps are described. The possibilities of mathematical tools for working with data dump files are examined. Method of traffic dumps analysis is obtained and its efficiency is checked. It has unlimited possibilities for parameters of any levels protocol extraction.

Keywords: traffic, semantic analysis, dump, computer networks, network protocols, network packet, mathematical tools, data extraction, data processing

References

1. Leland W. E., Taqu M. S., Willinger W., Wilson D. V. On the Self-Similar Nature of Ethernet Traffic. *Ref. Libr.* 1994. Vol. 2, N 1. P. 1—15.
2. MAWI Working Group Traffic Archive. URL: <http://mawi.wide.ad.jp/mawi> (data of access: 13.03.2014).
3. Tcdump/Libcap public repository. 2010—2014. URL: <http://www.tcpdump.org> (data of access: 13.03.2014).
4. Wireshark. URL: www.wireshark.org (data of access: 13.03.2014).
5. Network media specific capturing // Wireshark. URL: <http://wiki.wireshark.org/CaptureSetup/NetworkMedia> (data of access: 13.03.2014).
6. Traffic Trace Page. URL: <http://mawi.nezu.wide.ad.jp/mawi/samplepoint-F/2014/201401071400.dump.gz> (data of access: 13.03.2014).
7. Scilab — Open source software for numerical computation. URL: www.scilab.org (data of access: 13.03.2014).
8. GNU Octave // Operating system GNU. 1998—2013. URL: <https://www.gnu.org/software/octave> (data of access: 13.03.2014).
9. Sage: Open Source Mathematics Software. URL: www.sagemath.org (data of access: 13.03.2014).
10. Mathworks — MATLAB and Simulink for Technical Computing. 1994—2014. URL: www.mathworks.com (data of access: 13.03.2014).
11. Sharma N., Gobbert M. K. A comparative evaluation of MATLAB, octave, freemat, and scilab for research and teaching. Baltimore County: University of Maryland, 2010.
12. Sharma N. A comparative study of several numerical computational packages. // M. S. thesis. Baltimore County: University of Maryland, 2010.
13. Tracesplay. 1999—2008. URL: tracesplay.sourceforge.net (data of access: 13.03.2014).
14. Armenb/sharktools. *GitHub — Build software better, together.* URL: <https://github.com/armenb/sharktools> (data of access: 13.03.2014).
15. DisplayFilters. *Wireshark.* URL: <http://wiki.wireshark.org/DisplayFilters> (data of access: 10.04.2014).
16. System requirements OS Windows 7 — Microsoft Windows. *Microsoft Windows.* URL: <http://windows.microsoft.com/ru-ru/windows7/products/system-requirements> (data of access: 13.03.2014).
17. Installation Ubuntu Linux: Windows user instruction. *Russian-language documentation for Ubuntu.* URL: http://help.ubuntu.ru/wiki/ubuntu_install (data of access: 13.03.2014).
18. Web-Cite query result. *WebCite.* URL: <http://www.webcitation.org/64vYth5tU> (data of access: 13.03.2014).
19. The world's most popular free OS | Ubuntu. URL: www.ubuntu.com (data of access: 13.03.2014).

УДК 633.11; 321; 631.53

Ф. М. Велиев, аспирант, e-mail: asadzade@rambler.ru

Институт космических исследований природных ресурсов, г. Баку

Оптимизация вегетационного цикла роста растений по информационному критерию при использовании индекса листевой площади

Рассмотрена возможность информационной оптимизации вегетационного цикла роста растений при использовании индекса листевой площади. Предложена информационная модель индекса LAI с использованием элементов известной полуэмпирической модели CLAIR. Оптимизация предложенной информационной модели позволила синтезировать наиболее целесообразный характер взаимосвязи основных параметров известной модели CLAIR.

Ключевые слова: оптимизация, индекс листевой площади, вегетация, информационная модель, функционал, агрометеорология, оптическая толщина, рассеяние

Введение

Как сообщается в работе [1], в последние годы спутниковая информация дистанционного зондирования все чаще используется в высокоточных сельскохозяйственных технологиях. При этом наибольший интерес представляет индекс листовой площади (LAI), определяемый в качестве общей односторонней площади зеленых листьев, попадающей на единицу площади поверхности исследуемой территории. Индекс LAI широко используется в качестве агрономического индикатора в различных областях сельскохозяйственного растениеводства. Например, карты индекса LAI используются в агрометеорологических моделях для изучения потребности растительности во влаге, для прогнозирования масштабов будущего урожая. Для практической оценки индекса LAI используют как полуэмпирический/статистический подход, так и физические модели с использованием теории переноса излучения. Эмпирические/статистические методы предполагают вычисление вегетационных индексов, проведение полевых измерений, исследование регрессионных зависимостей и т.д. Второй подход предполагает усовершенствование физических моделей переноса излучения, осуществление радиометрической и атмосферной коррекции, проведение синхронных атмосферных измерений (оптической толщины аэрозоля, водяных паров и т.д.). В работе [1] приведены результаты исследований по оценке индекса LAI с использованием снимков спутников DEIMOS-1, при анализе которых была использована полуэмпирическая модель CLAIR [2, 3].

В настоящей статье рассмотрена возможность разработки информационной модели индекса LAI на базе известной модели CLAIR и оптимизации этой модели для нахождения оптимальной взаимосвязи между основными параметрами модели CLAIR.

Постановка задачи

Модель CLAIR основана на инверсном экспоненциальном соотношении между LAI и водяным дифференциальным вегетационным индексом ($WDVI$), определяемым следующим образом

$$WDVI = \rho_{nir} - \rho_{red} \frac{\rho_{s\ nir}}{\rho_{s\ red}}, \quad (1)$$

где $\frac{\rho_{s\ nir}}{\rho_{s\ red}}$ — крутизна "почвенной" линии в диаграмме "NIR—RED"; ρ_{red} — отраженный сигнал вегетации в красной спектральной зоне; ρ_{nir} — отраженный сигнал вегетации в ближней инфракрасной зоне; сокращение "nir" означает ближнюю инфракрасную зону; "red" означает красную спектральную зону.

В формуле (1) учет характеристики "почвенной" линии объясняется необходимостью коррекции влияния почвенного фона при зондировании растительного участка. Согласно [2, 3] индекс LAI находится в следующей связи с показателем $WDVI$

$$LAI = -\frac{1}{\alpha} \ln \left(1 - \frac{WDVI}{WDVI_{\infty}} \right), \quad (2)$$

где α — коэффициент экстинкции и рассеяния; $WDVI_{\infty}$ — асимптотическое значение $WDVI$.

Значения $WDVI_{\infty}$ и α определяются эмпирическим путем. По существующим научным представлениям α и $WDVI_{\infty}$ для каждого вида растений при равных агрометеоусловиях являются постоянными величинами [2—6]. При этом в работах [4, 6] показатель α характеризуется в качестве скорости приближения $WDVI$ к своему асимптотическому значению.

В настоящей статье ставится задача разработки информационной модели индекса LAI на базе известной модели CLAIR, оптимизация этой модели и выяснение ряда несоответствий в интерпретации показателей α и $WDVI_{\infty}$.

Прежде всего отметим, что априорный тезис о постоянстве α не подтверждается результатами экспериментальных исследований разных авторов. Так, согласно [6] в вегетативной стадии развития растительности $\alpha = 0,252$, а в генеративной стадии $\alpha = 0,530$.

Согласно [1] вычисленные на базе четырех космических снимков идентичных вегетационных полей значения этого показателя составили $0,34 \pm 0,02$; $0,35 \pm 0,01$; $0,37 \pm 0,01$; $0,34 \pm 0,02$.

Необходимо отметить, что определение α в качестве постоянной величины, с одной стороны, и в качестве скорости приближения индекса $WDVI$ к своему асимптотическому значению, с другой стороны, некорректно также с математической точки зрения, поскольку функция в принципе не может приближаться к своей асимптоте с постоянной скоростью, так как в этом случае она достигла бы свою асимптоту. Таким образом, изменчивость α в течение вегетационного цикла развития растений является доказанным фактом как экспериментально, так и из математических соображений. С учетом вышесказанного имеет смысл ввести на рассмотрение функцию

$$WDVI = f(\beta), \quad (3)$$

где

$$\beta = \frac{1}{\alpha}, \quad (4)$$

и исследовать динамику изменения этой функции с точки зрения выбранного функционала цели, характеризующего состояние здоровья растений.

Решение задачи

Для решения вышеуказанной задачи далее излагается предлагаемая информационная модель LAI. Основу предлагаемой информационной модели составляет выражение (2) которую запишем в следующем виде:

$$LAI = \beta \log_2 \left[\frac{1}{1 - \frac{f(\beta)}{WDVI_\infty}} \right], \quad (5)$$

где

$$WDVI = f(\beta), \quad (6)$$

$$\beta = \frac{1}{\alpha \log_2 e}. \quad (7)$$

Составим интегральный информационный функционал (8), где показатель β условно характеризует число элементов информационного сообщения;

$\frac{1}{1 - \frac{f(\beta)}{WDVI_\infty}}$ характеризует число достижимых гра-
дационных уровней

$$F = \int_0^{\beta_m} \beta \log_2 \left[\frac{1}{1 - \frac{f(\beta)}{WDVI_\infty}} \right] d\beta, \quad (8)$$

где β_m — максимальное значение показателя β .

Для конкретного вида растений и фиксированных агрометеорологических условий примем следующее ограничительное условие

$$\int_0^{\beta_m} f(\beta) d\beta = C. \quad (9)$$

С учетом выражений (8) и (9) составим полный безусловный информационный функционал

$$F = \int_0^{\beta_m} [\beta, f(\beta), WDVI_\infty, \gamma] d\beta = \int_0^{\beta_m} \beta \log_2 \left[\frac{1}{1 - \frac{f(\beta)}{WDVI_\infty}} \right] d\beta + \gamma \int_0^{\beta_m} f(\beta) d\beta, \quad (10)$$

где γ — множитель Лагранжа.

Согласно методу Эйлера решение оптимизационной задачи должно удовлетворять требованию

$$\frac{dM[\beta, f(\beta), WDVI_\infty, \gamma]}{df(\beta)} = 0, \quad (11)$$

где

$$M = \beta \log_2 \left[\frac{1}{1 - \frac{f(\beta)}{WDVI_\infty}} \right] + \gamma f(\beta).$$

С учетом выражений (10) и (11) получим

$$\frac{\beta}{\ln 2 [WDVI_\infty - f(\beta)]} + \gamma = 0. \quad (12)$$

С учетом выражений (9) и (11) нетрудно получить следующее выражение:

$$\gamma = \frac{\beta_m^2}{2 \ln 2 (C - \beta_m WDVI_\infty)}. \quad (13)$$

С учетом выражений (12) и (13) окончательно получаем:

$$f(\beta) = WDVI_\infty + \frac{2\alpha_m^2 \left(C - \frac{WDVI_\infty}{\alpha_m} \right)}{\alpha}, \quad (14)$$

где α_m — максимальное значение α .

Для проверки характера выявленного экстремума получим производную выражения (12) по $f(\beta)$. Имеем

$$\frac{d^2 M}{dM^2} = \frac{\beta}{\ln 2 [WDVI_\infty - f(\beta)]^2}. \quad (15)$$

Как видно из выражения (15), вторая производная M является положительной величиной, что указывает на то, что при решении (14) информационная модель (8) или интегральное значение LAI, обозначенное как LAI_{in} , где

$$LAI_{in} = \int_0^{\beta_m} (LAI) d\beta, \quad (16)$$

достигает минимальной величины. Следовательно, с точки зрения достижения наибольшей продуктивности и здоровья сельскохозяйственных растений зависимость в виде (14) является наилучшим вариантом. С учетом очевидного неравенства

$$\frac{WDVI_\infty}{\alpha_m} > C \quad (17)$$

оптимальным оказывается такая взаимосвязь между $WDVI$ и α , при которой рост $WDVI$ сопровождается уменьшением α .

Заклучение

Рассмотрена возможность информационной оптимизации вегетационного цикла роста растений при использовании индекса листевой площади.

Предложена информационная модель индекса LAI с использованием элементов известной полуэмпирической модели CLAIR.

Оптимизация предложенной информационной модели позволила синтезировать наиболее целесообразный характер взаимосвязи основных параметров известной модели CLAIR.

Список литературы

1. Vuolo F., Neugebauer N., Bolognesi S. F., Atzberger C., D'Urso G. Estimation of leaf area index using DEIMOS-1 data: Re-

lationship between two agricultural areas // *Remote Sensing*. 2013. N. 5. P. 1274–1291; doi:10.3390/rs5031274.

2. Clevers J. G. P. W. Application of a weighted infrared-red vegetation index for estimating leaf area index by correcting for soil moisture // *Remote Sens. Environ.* 1989. V. 29. P. 25–37.

3. Clevers J. G. P. W. The derivation of a simplified reflectance model for the estimation of leaf area index // *Remote Sens. Environ.* 1988. V. 25. P. 53–69.

4. Clevers J. G. P. W., van Leeuwen H. J. C. Combined use of optical and microwave remote sensing data for crop growth monitoring // *Remote Sens. Environ.* 1996. V. 56. P. 42–51.

5. Mengesha T., Kooustra L., de Bruin S., Milla R. Z., Schaepman M. Methodology comparison of quantitative LAI retrieval using imaging spectroscopy and GEO-spatial interrelation in a soft wood forest // *Proceeding of 4th EARSel Workshop on Imaging. New quality in environmental studies*. Warsaw. 2005. P. 641–651.

6. Clevers J. G. P. W., Vonder O. W., Jongschaap R. E. E., Desprats J. F., King C., Prevot L., Bruguier N. A Semi-empirical approach for estimating plant parameters within the reseda-project // *International Archives of Photogrammetry and Remote Sensing*, Amsterdam. 2000. V. XXXIII. P. B7. P. 272–279.

F. M. Veliyev, Graduate Student, e-mail: asadzade@rambler.ru
Institute of space researches of natural resources, Baku

Optimization of Development Stage of Vegetation on Information Criterion Using the Leaf Area Index

The last years the satellite remote sensing information is widely used in high accuracy agricultural technologies. The most interesting parameter of vegetation is the Leaf Area Index (LAI) which is determined as total one-sided area of green leaves located at the area unit of studied territory. LAI is widely used as an agronomical indicator in various fields of agricultural vegetation growth.

For practical assessment of LAI both the semi empirical/statistical approaches and the physical models based on radiation transfer theory are used. In the present article the feasibility of development of information model of LAI on the basis of known model of CLAIR is considered and the optimization of this model for finding up of optimal interrelation between major parameters of CLAIR is carried out.

The aim of this article is development of information model of LAI on the basis of known CLAIR model, optimization of former and clearing of some irregularities in interpretation of several related parameters.

The information model of LAI (Leaf Area Index) which use the elements of known semi-empirical CLAIR model is suggested.

Optimization of suggested information model allows to synthesize the expedient character of interrelation of major parameters of CLAIR.

Keywords: optimization, LAI (Leaf Area Index), vegetation, information model, functional, agrometeorology, optical depth, scattering

References

1. Vuolo F., Neugebauer N., Bolognesi S. F., Atzberger C., D'Urso G. Estimation of leaf area index using DEIMOS-1 data: Relationship between two agricultural areas. *Remote Sensing*. 2013. N. 5. P. 1274–1291; doi:10.3390/rs5031274.

2. Clevers J. G. P. W. Application of a weighted infrared-red vegetation index for estimating leaf area index by correcting for soil moisture. *Remote Sens. Environ.* 1989. V. 29. P. 25–37.

3. Clevers J.G.P.W. The derivation of a simplified reflectance model for the estimation of leaf area index. *Remote Sens. Environ.* 1988. V. 25. P. 53–69.

4. Clevers J. G. P. W., van Leeuwen H. J. C. Combined use of optical and microwave remote sensing data for crop growth monitoring. *Remote Sens. Environ.* 1996. V. 56. P. 42–51.

5. Mengesha T., Kooustra L., de Bruin S., Milla R. Z., Schaepman M. Methodology comparison of quantitative LAI retrieval using imaging spectroscopy and GEO-spatial interrelation in a soft wood forest. *Proceeding of 4th EARSel Workshop on Imaging. New quality in environmental studies*. Warsaw. 2005. P. 641–651.

6. Clevers J. G. P. W., Vonder O. W., Jongschaap R. E. E., Desprats J. F., King C., Prevot L., Bruguier N. A Semi-empirical approach for estimating plant parameters within the reseda-project. *International Archives of Photogrammetry and Remote Sensing*. Amsterdam. 2000. V. XXXIII. P. B7. P. 272–279.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ MODELING AND OPTIMIZATION

УДК 004.42

Н. М. Якимов, аспирант, **С. Н. Чувашев**, д-р физ.-мат. наук, проф., root@livid.pp.ru, snchuv@mail.ru, МАТИ—РГТУ имени К. Э. Циолковского

Программное средство для комплексного математического моделирования сложных технических объектов

Предложен новый программный комплекс для моделирования сложных технических объектов. Приведено сравнение с существующими решениями (MATLAB Simulink, OpenModelica, MapleSim и др.). Предложенный программный комплекс, по-видимому, превосходит их по совокупности критериев, выработанных в результате опыта создания моделей различных объектов. Описан пример применения комплекса (для механической мельницы).

Ключевые слова: математическое моделирование, комплексные модели, программный комплекс, блочное моделирование, компонентное моделирование

Введение

Быстрый рост вычислительной мощности ЭВМ в последние десятилетия делает возможным использование компьютерного математического моделирования для все более сложных объектов. Математическое моделирование позволяет избежать многократного использования прототипов и при правильном подходе дает возможность сократить стоимость разработки концептуально новых решений и продуктов.

При моделировании сложных технических объектов важной оказывается возможность изолированного исследования подсистем, составляющих данный объект. Это связано как с необходимостью верификации построенных математических моделей, отладки построенных программных комплексов, так и с необходимостью решения задач различных дисциплин в различных подсистемах. Изолированное исследование подсистем технически сложного объекта позволяет, с одной стороны, верифицировать и отлаживать более простые системы и затем идти от простого к сложному, а с другой стороны, упрощает разработку, параллельно ведущуюся различными группами исследователей. Модель объекта, составленную из моделей подсистем, можно представить как некий комплекс моделей или как комплексную модель. Дополнительным преимуществом комплексных моделей является возможность повторного использования уже разработанных моделей подсистем в комплексных моделях других объектов, т. е. использования некой библиотеки моделей при разработке.

При моделировании технически сложных объектов возникают вычислительные задачи большой

размерности, для решения которых необходимо применять различные специализированные математические методы. Для уменьшения размерности задачи можно использовать также подходы иерархического моделирования и табличного задания процессов на основе экспериментальных или модельных результатов.

Использование библиотеки моделей накладывает требование возможности быстрой адаптации библиотечных моделей под условия разрабатываемой модели.

На основе вышесказанного, а также нашего многолетнего опыта разработки комплексных моделей различных объектов (космических энергоустановок [1], гиперзвуковых летательных аппаратов [2, 3], судов на воздушной подушке [4] и др.) можно сформулировать следующие требования для программного комплекса, решающего задачу комплексного моделирования:

- автоматическая сборка комплексной модели из моделей подсистем;
 - возможность использования различных численных методик для решения задач высокой размерности (в том числе поддержка пользовательских компонентов-решателей);
 - поддержка подходов иерархического моделирования и табличного задания процессов;
 - возможность быстрой корректировки библиотечных моделей под изменяющиеся требования.
- С точки зрения удобства использования программного пакета можно сформулировать несколько дополнительных требований:
- графическое редактирование связей между моделями подсистем;

- использование стандартного языка для разработки моделей;
- низкая стоимость лицензии на одно рабочее место.

Графическое редактирование является наиболее наглядным представлением, что дает более высокую устойчивость к ошибкам. Использование стандартного языка уменьшает порог вхождения и затраты на обучение. Последнее требование продиктовано необходимостью возможности параллельной разработки различными группами исследователей: при высокой стоимости лицензии на рабочее место необходимы либо крупные капиталовложения для обеспечения каждой группы, либо дополнительные программные и технические средства для обеспечения параллельного или удаленного доступа к программному комплексу.

Анализ существующих решений

На текущий момент на рынке программного обеспечения существует ряд решений для блочного и компонентного моделирования систем. Многие из этих решений на данный момент активно развиваются. Тем не менее, большинство из существующих решений оптимизировано под определенные классы задач, в связи с чем их применение к более широкому кругу проблем осложнено.

Наиболее известными сейчас являются решения для блочного моделирования, такие как MATLAB Simulink [5], VisSim [6], и их открытые аналоги, такие как Scicos [7]. Общей проблемой подобных решений является их ориентированность на разработку моделей для обработки сигналов, в частности, электрических и гидравлических цепей. Несмотря на то, что для подобных применений эти решения подходят достаточно хорошо, для реализации более сложных моделей требуются значительные усилия [8].

Другой проблемой этих систем часто является жестко заданный набор решателей дифференциальных уравнений. В случае систем с открытым исходным кодом эта проблема стоит не столь остро, однако если поставляемые решения не удовлетворяют требованиям решаемой задачи, добавление пользовательских решателей в открытые системы сопряжено с большим объемом работ по освоению существующей кодовой базы [9].

Сравнительно новой альтернативой подходу блочного моделирования является компонентное моделирование систем. К программным решениям этого класса относятся, например, EcosimPro [10], ASCEND [11], EMSO [12] и различные реализации стандарта Modelica [13], как открытые, например, OpenModelica [14] и JModelica [15], так и проприетарные, например, Dymola [16], MapleSim [17], AMESim [18] и др.

Все перечисленные решения для компонентного моделирования обладают общей проблемой: адаптация старых исходных кодов моделей на

Fortran, C или C++ требует генерации специальным образом сформированной динамической библиотеки (DLL или SO) либо специальным образом сформированного исходного кода (в случае AMESim).

Другой проблемой, вытекающей из предыдущей, является необходимость освоения собственного скриптового языка данной системы. При этом даже решения на основе Modelica обладают присущими только им особенностями языка, которые необходимо знать для эффективного использования этих решений.

Подключение сторонних решателей и/или интеграторов, как и в случае с решениями для блочного моделирования, также поддерживается не всеми перечисленными программными пакетами. Подробная документация приведена только для ASCEND (в котором на момент написания эта функциональность является экспериментальной) и EMSO.

К недостаткам закрытых решений также можно отнести их высокую стоимость: например, Dymola (не продается в России) оценивается в \$5000 для полной редакции (\$1200 для упрощенной), а дополнительные компоненты, такие как 3D-визуализация, дополнительно увеличивают стоимость на \$1200...1500. AMESim оценивается в \$11 000, MapleSim — около \$7000. Все цены приведены для лицензий на одного пользователя.

Что касается открытых решений, ASCEND, несмотря на длительное присутствие на рынке, на момент написания значительно проигрывает другим программным пакетам по функциональности (которая либо не реализована, либо является экспериментальной). Это позволяет предположить, что разработка ведется крайне медленными темпами. Подобное заключение можно сделать и о "полузакрытом" EMSO, активная разработка которого не ведется с 2010 г., а многие цели, которые планировалось реализовать еще к 2008 г., не реализованы до сих пор. К тому же по условиям лицензии основной программный пакет EMSO доступен только для образовательных и некоммерческих применений.

JModelica несмотря на активную разработку на момент написания реализует стандарт Modelica только частично [19], поэтому использование может быть дополнительно осложнено.

Более полное сравнение программных продуктов приведено в таблице.

Предлагаемое программное решение

Из вышесказанного следует, что использование существующих программных продуктов для создания и анализа комплексных моделей не всегда удобно. В связи с этим был разработан новый программный пакет, смягчающий многие из перечисленных недостатков.

Автоматическая сборка комплексной модели из моделей подсистем обеспечивается модулем "ком-

Сравнение программных продуктов

	Simulink	VisSim	Scicos	EcosimPro	ASCEND	EMSO	OpenModelica	JModelica	Dymola	SM1	MapleSim	AmeSim
1	+	+	+	+	+	+	+	—	+	+	+	+
2	—	+	*	—	+	+	*	*	—	—	—	—
3	—	—	*	—	+	+	*	*	—	—	—	—
4	†	†	†	†	†	†	†	†	†	†	†	†
5	†	†	†	+	+	+	†	—	+	†	†	+
6	—	—	—	—	—	—	‡	‡	‡	‡	‡	‡
7	—	—	—	+	+	+	+	+	+	+	+	+
8	+	+	+	+	—	+	+	—	+	+	+	+
9	W,L,M	W	W,L,M	W	W,L	W,L	W,L,M	W,L,M	W,L	W,M	W,L,M	W,L
10	\$5000	\$3840	Беспл.	\$1500—2500	Беспл.	Некомм.	Беспл.	Беспл.	\$5000	\$2450	\$7000	\$11 000

Примечание. 1 — автоматическая сборка комплексной модели из моделей подсистем; 2 — пользовательские решатели; 3 — пользовательские интеграторы; 4 — табличное задание процессов/параметров; 5 — сравнительно легкая адаптация "старых" моделей на компилируемых языках (Fortran, C, etc); 6 — использование стандартного языка для разработки моделей; 7 — быстрая корректировка библиотечных моделей; 8 — графическое редактирование связей между моделями подсистем; 9 — поддерживаемые ОС (W — Windows, L — Linux, M — Mac OSX); 10 — примерная стоимость лицензии на рабочее место.

* — функция теоретически доступна, но не поддерживается. Требуется редактирование исходного кода или дополнительные технические приемы.

† — функция, реализуемая посредством сложных манипуляций другими возможностями программы.

‡ — использование узко специфичного языка Modelica.

пилятора", принимающим на вход файл-описание и исходные коды и создающим двоичный исполняемый файл комплексной модели, который затем можно использовать отдельно от программного комплекса. Модуль "компилятора" написан с использованием C++ и пакета библиотек Boost.

Модульная архитектура позволяет использование пользовательских интеграторов и решателей. Стандартный API требует от модуля решателя только использование соглашений на именование и прототипы функций и использование фиксированного XML-формата для управления настройками решателя и модели.

Использование полного C++ в качестве языка описания моделей позволяет легко вставлять в модели таблично задаваемые параметры с применением любого метода интерполяции.

Адаптация "старых" моделей, как правило, сводится к написанию простого "адаптера", описывающего входы и выходы, и небольшой модификации исходной модели для работы с "адаптером".

Библиотечные модели целиком копируются в проект, поэтому их изменение в рамках проекта не требует дополнительных действий. Возможна модификация моделей на этапе связывания в комплексную модель.

Инструмент графического редактирования связей между моделями подсистем написан на C++ с использованием графических библиотек Qt.

Все основные инструменты программного пакета написаны на C++ с использованием кросс-платформенных библиотек, что позволяет легко переносить программный пакет на любую платформу с поддержкой этих библиотек, в частности на Linux, Windows и Mac OSX.

Структура комплексной модели

Математическую модель можно представить как некую систему уравнений. Система уравнений характеризуется, в свою очередь, переменными, значения которых могут быть постоянны либо изменяться дискретно или непрерывно, и уравнениями, описывающими связь этих переменных между собой. Для разрешимости задачи математического моделирования необходимо, чтобы система, описывающая данную модель, была замкнутой. Соответственно, если исходная система не замкнута, для введения дополнительных условий необходимо добавить к исходной еще одну или несколько систем уравнений, описывающих поведение неопределенных в исходной системе переменных. Рассуждая таким образом, можно предложить следующую схему представления моделей подсистем: моделью подсистемы ("простой" моделью) является некоторая система уравнений, имеющая N неопределенных переменных, где $N \geq 0$. Совокупность простых моделей можно считать комплексной моделью, причем комплексная модель также может быть незамкнутой. Окончательно комплексная модель, пригодная для моделирования, может представлять такую совокупность простых и комплексных моделей, в которой все переменные определены. Переменные, определенные в рамках данной модели, мы будем называть "выходными параметрами". Переменные, не определенные в рамках данной модели, будем называть "входными параметрами".

Для описания комплексной модели используется иерархическое дерево: простые модели, описываемые с помощью исходного кода на языке C++ и файла XML с описанием входных и выходных параметров, являются "листьями" этого дерева. "Листья" объединяются в "ветви", представляющие комплексные модели. Комплексные модели должны

дополнительно описывать связи параметров между входящими в них моделями. Наконец, "корнем" дерева является комплексная модель, состоящая из простых и комплексных моделей, в которой все входные параметры определены. Все дерево далее будем называть "проектом".

В качестве формата файлов проекта выбраны zip-файлы со сжатием deflate, содержащие исходные коды входящих в проект моделей, и XML-файл описания, содержащий информацию о входных и выходных параметрах моделей (в том числе название, описание, размерность, и т. п.), а также о связях между параметрами и исходном коде. Кроме того, отдельные модели при необходимости сохраняются в аналогичных по структуре zip-файлах и могут быть впоследствии загружены в другой проект.

Модуль "компилятора"

Модуль "компилятора" преобразует файлы проекта в бинарные исполняемые файлы модели, пригодные для расчетов. Этот процесс происходит в несколько этапов:

1. Строится граф зависимостей всех параметров модели.
2. Проверяется наличие циклических зависимостей.
3. Исходные коды моделей переписываются в вид, пригодный для использования с модулями решателей.
4. Определяется порядок расчета параметров комплексной модели на основе топологической сортировки.
5. Генерируется функция состояния системы, пригодная для использования с модулями решателей.
6. Полученные коды компилируются и линкуются с решателем с помощью внешнего компилятора C++.
7. Генерируется файл настроек.

Граф зависимостей параметров строится на основе синтаксического анализа исходных кодов моделей и связей, указанных в XML-описании проекта. Для проведения синтаксического анализа используются регулярные выражения (библиотека Boost Xpressive).

Поиск циклических зависимостей и топологическая сортировка проводятся стандартными алгоритмами.

На третьем этапе все параметры линеаризуются, т. е. представляются в виде массива скаляров, и соответствующие именованные параметры моделей заменяются на внутреннее представление. Кроме этого, во избежание конфликта имен функций функции моделей переименовываются уникальным образом (добавлением префикса, характеризующего положение модели в иерархии проекта). Входные и выходные параметры, связанные равенством, используют одну ячейку памяти, однако с разным спецификатором доступа: изменение входных параметров запрещено.

Функция состояния системы для случая моделей, описываемых системами обыкновенных дифференциальных уравнений, возвращает правую часть задачи Коши. Для случая систем дифференциально-алгебраических уравнений или алгебраических уравнений — вектор остаточных членов системы.

Файл настроек представляет собой файл XML, содержащий информацию о начальных значениях параметров, задаваемых дифференциальными уравнениями, значениях параметров, обозначенных как постоянные, и настройках модуля решателя.

Инструмент графического редактирования связей

Инструмент графического редактирования связей предназначен для связывания отдельных "простых" моделей в комплексную. Модель графически представляется в виде прямоугольника с выходящими из его ребер линиями с квадратом на конце (далее "блок"), отражающими входные и выходные параметры модели (далее "коннекторы"). При этом "входные" коннекторы (для входных параметров) расположены на правой грани, "интегрируемые" — на верхней, постоянные — на нижней и "выходные" — на правой (рис. 1, 2, см. вторую сторону обложки).

Методом "drag&drop" возможно связывание всех прочих коннекторов с входными. Для упрощения визуального восприятия коннекторы автоматически кодируются цветом. Цвета связующей линии и входного коннектора соответствуют цвету связанного выходного коннектора. Возможно связывание одного выходного коннектора с несколькими входными, но не наоборот.

Также возможно создание блоков комплексных моделей, включающих в себя несколько "простых" или других комплексных моделей, как полностью определенных, так и частично. При этом не определенные в рамках такого блока параметры должны быть "экспортированы" (экспорт реализуется пунктом контекстного меню коннектора). На блоке комплексной модели отображаются только коннекторы "экспортированных" параметров. Блок комплексной модели можно "развернуть" двойным кликом на нем.

Любой блок может быть сохранен в виде файла модели. Также любой ранее сохраненный блок возможно загрузить в новую модель.

Кроме того, инструмент графического редактирования связей включает в себя возможности для редактирования исходного кода моделей, добавления или удаления параметров и возможности создания новых моделей в процессе работы.

Модуль решателя

Модуль решателя представляет собой статическую библиотеку со строго определенным API. Строго говоря, API определяет всего одну функцию solve с одним параметром — экземпляром вспомогательного класса, предоставляющим информацию о времени счета, времени остановки, настройках

решателя, а также вспомогательные функции, реализующие инициализацию, вывод и расчет правых частей задачи Коши.

На текущий момент реализованы следующие решатели:

- DOPRI5[20] (адаптирован из старого кода на языке FORTRAN);
- RADAU5[21] (адаптирован из старого кода на языке FORTRAN);
- SUNDIALS CVODE [22] (написан интерфейс, "стыкующий" библиотеки SUNDIALS с API решателя).

Инструмент анализа решения

Инструмент анализа решения позволяет изменять начальные условия и постоянные без повторной сборки модели, графически отображает решение, позволяет сохранять и конвертировать файл решения в различные форматы (на текущий момент — `ascii`-таблица и `teerplot`).

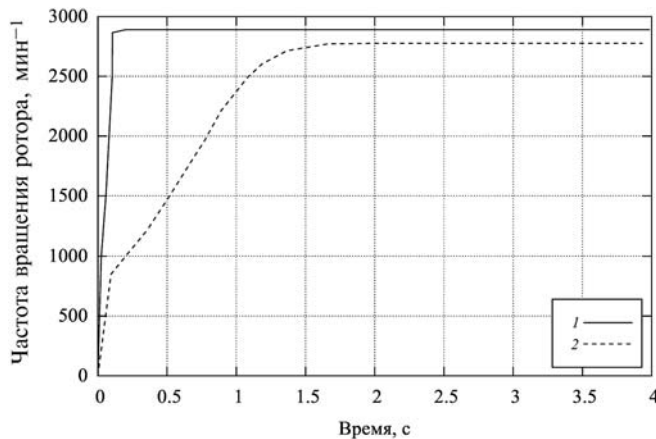


Рис. 3. Частота вращения ротора от времени при различных моментах инерции ротора:

1 — радиус ротора 1,5 см, момент инерции $3,1 \cdot 10^{-5}$ кг · м²; 2 — радиус ротора 3,0 см, момент инерции $5,0 \cdot 10^{-4}$ кг · м²

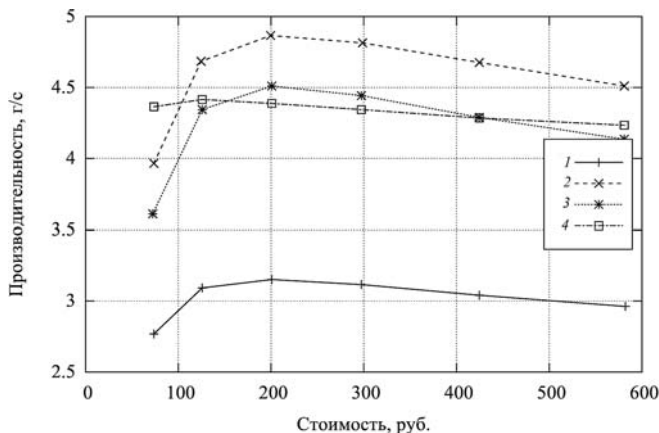


Рис. 4. График производительности от стоимости для различных размеров рабочей камеры:

1 — радиус рабочей камеры $R = 4,4$ см; 2 — $R = 5,5$ см; 3 — $R = 6,6$ см; 4 — $R = 7,7$ см

Формат решения предполагает текстовое описание, которое также вводится из интерфейса данного инструмента.

В процессе решения графики периодически обновляются (рис. 3). Также отображается общий прогресс решения.

Возможен многопараметрический анализ, реализуемый многократным расчетом модели (рис. 4).

Пример использования

Рассматривается пример использования предлагаемого программного средства для моделирования технически сложных объектов на примере мельницы для измельчения твердых органических материалов. С одной стороны, подобное устройство годится как пример системы, принцип действия которой знаком из бытовой техники, с другой стороны, его моделирование и оптимизация потребовались, в частности, для решения задачи экономически эффективного получения микронных порошков органического сырья, критически важной для масштабной программы получения биотоплива.

Структурная схема комплексной модели показана на рис. 1 (см. вторую сторону обложки). Комплексная модель мельницы состоит из блоков:

- модели асинхронного однофазного электрического двигателя;
- модели ременной передачи;
- модели источника синусообразного тока;
- модели рабочей камеры.

Модель электрического двигателя разработана в нашей работе [1] на основе работ [23, 24]. В модели учтены эффекты влияния ширины зазора, взаимовлияние катушек ротора и статора, индуктивности рассеяния катушек, описывается динамика раскрутки и торможения ротора, динамические эффекты на стартовом конденсаторе. Оценивается стоимость двигателя (пропорционально объему материалов, требуемых для его изготовления, что верно при массовом производстве).

Модель помола построена на основе предположения, что обмен импульсом между частицами эффективен, т. е. что частицы движутся с минимальными скоростями относительно друг друга. Тогда скорость частиц v направлена тангенциально, причем ее модуль пропорционален радиальному размеру r :

$$v = \omega r.$$

Частота вращения ω определяется уравнением баланса моментов для вращения [25]

$$J \frac{d\omega}{dt} = M_i - M_w.$$

Здесь J — момент инерции; t — время; M_i — вращательный момент от ударного взаимодействия частиц с ножом; M_w — вращательный момент от трения частиц о стенку. Для цилиндрической засыпки

$$J = \int_0^R \rho r^2 dV = \int_0^R \rho r^2 h 2\pi r dr = 2\pi \rho h R^4 / 4,$$

где V — объем; h — высота; $\rho = C_p \rho_0$ — плотность засыпки; C_p — объемная доля твердого материала, для сферических частиц $C_p = \pi/6$, ρ_0 — плотность твердого материала; R — радиальный размер камеры.

При трении частиц о стенку тормозящий момент

$$M_w = C_w p_w (2\pi R h) R,$$

где C_w — коэффициент трения; p_w — давление на стенку, для цилиндрической засыпки

$$p_w = \int_0^R a \rho dr = \int_0^R r \omega^2 \rho dr = R^2 \omega^2 \rho.$$

Здесь a — ускорение.

Вращательный момент от ударного взаимодействия частиц с двумя ножами

$$\begin{aligned} M_i &= 2R_k v_k (C_k + 1) 2m(v_k - v_m) = \\ &= 2R_k (C_k + 1) 2\rho h_k \delta_k (R_k \Omega - R_k \omega) (R_k \Omega - R_k \omega) = \\ &= 2(C_k + 1) 2R_k^3 \rho h_k \delta_k (\Omega - \omega)^2, \end{aligned}$$

где R_k — радиальная координата рабочей поверхности ножа; h_k — ее высота; δ_k — толщина; m — масса частицы; v_k — частота соударений; v_k — скорость ножа; v_m — скорость частицы до соударения; Ω — частота вращения ножа; C_k — коэффициент упругости удара; $C_k = 1$ для абсолютно упругого и $C_k = 0$ для абсолютно неупругого удара.

Помол при высокоскоростном ударе рассчитывался из уравнения [26]

$$\partial N_p(t)/\partial t = N_{p0} K v_{rel}^2 / x_p^{1/2},$$

где v_{rel} — относительная скорость столкновения частицы и ножа; t — время; $N_p(t)$ — текущее, N_{p0} — начальное число частиц; x_p — текущий средний характерный размер частиц, связанный с $N(t)$; K — коэффициент, определяемый механическими свойствами обрабатываемого материала.

Каждый из блоков разрабатывался и тестировался отдельно. Полные исходные коды модели составляют 170 строк. Исходные коды, сгенерированные компилятором, составляют 809 строк.

На рис. 2 (см. вторую сторону обложки) показана структурная схема комплексной модели электрического двигателя. Модель двигателя можно изолированно использовать, например, для анализа раскрутки ротора в режиме холостого хода. Так, например, при увеличении момента инерции ротора (за счет увеличения радиуса) время раскрутки увеличивается (см. рис. 3).

При использовании модели электродвигателя в составе модели мельницы отдельные модели автоматически объединяются в общий комплекс. С помощью этого комплекса можно проводить анализ и оптимизацию различных характерных параметров конструкции и режимов, например, в терминах стоимость—производительность. Под производи-

тельностью понимается масса измельченного материала, отнесенная ко времени измельчения.

В качестве примера рассмотрим задачу оптимизации соотношения производительность/стоимость по параметру размера рабочей камеры. Проводится несколько серий расчетов работы мельницы при пропорциональном изменении размеров элементов двигателя. Для каждого набора параметров определяется время достижения необходимого размера частиц (300 мкм). По заданной массе материала определяется производительность. По геометрическим параметрам определяется стоимость. Строится график производительности от стоимости для различных размеров рабочей камеры (рис. 4).

С помощью комплексной модели данного технически сложного объекта продемонстрирована принципиальная возможность использования ее на стадии эскизного проектирования мельницы в целях повышения качества проектирования и снижения издержек на разработку. При этом моделируются аспекты различных рабочих процессов, преодолеваются трудности, связанные с различными пространственными и временными масштабами рабочих процессов в различных подсистемах. Продемонстрирована принципиальная возможность проведения многопараметрического анализа на этапе эскизного проектирования. Показано значительное уменьшение объема необходимого исходного кода (по крайней мере, в 4 раза).

Заключение

Разработано программное обеспечение, которое, по-видимому, лучше других удовлетворяет комплексу требований для математического моделирования технически сложных объектов, состоящих из подсистем, рабочие процессы в которых относятся к различным областям знаний и моделируются различными автономными группами исследователей с различной степенью подробности (в том числе с использованием подходов иерархического моделирования и табличного задания процессов). Язык разработки автономных моделей подсистем общеизвестный — C++, переносимый на большинство популярных платформ, что снижает порог вхождения новых групп исследователей.

Список литературы

1. **Рабочие** процессы в узлах турбокомпрессорного генератора (компрессор, турбина, электрогенератор) / С. Н. Чувашев, И. Б. Пьянков, Е. С. Чувашева, Н. М. Якимов, И. Г. Зорина. М.: Перо, 2011. 550 с.
2. **Чувашева Е. С., Чувашев С. Н., Зорина И. Г.** Комплексная математическая модель для концептуального проектирования высокоскоростных летательных аппаратов // Информационные технологии. 2012. № 11(195). С. 10—14.
3. **Чувашева Е. С., Чувашев С. Н.** Выбор рациональных характеристик высокоскоростных летательных аппаратов разных масштабов на основе комплексной математической модели // Информационные технологии. 2013. № 8. С. 12—16.
4. **Полов С. Д., Чувашев С. Н.** Разработка технологии выбора несущего комплекса для транспортных средств на воздушной по-

душке (ТСВП), предназначенных для эксплуатации на Севере и в Сибири // Сб. статей международной научно-практической конференции Инновационное развитие современной науки / Вод. А. А. Сукиасян. Уфа: РИЦ БашГУ. 2014. Т. 3. С. 287–295.

5. **Simulink**, User's Guide, Version 7. The MathWorks Inc, 2009.
6. **VISSIM** User Manual, Version 4.00. PTV AG, Germany, 2004.
7. **Nikoukhah R., Steer S.** SCICOS — a dynamic system builder and simulator // *Computer-Aided Control System Design*, 1996. Proceedings of the 1996 IEEE International Symposium on. IEEE, 1996. P. 430–435.
8. **Nikoukhah M. N. R.** Modeling and simulation of differential equations in Scicos // *Modelica Conference*. 2006.
9. **Scicos** FAQ. URL: <http://www.scicos.org/FAQ.html#D2>
10. **EcosimPro**: User Manual. EA International, 2004.
11. **ASCEND**: An object-oriented computer environment for modeling and analysis: The modeling language / P. C. Piela, T. G. Epperly, K. M. Westerberg, A. W. Westerberg // *Computers & chemical engineering*. 1991. V. 15, N 1. P. 53–72.
12. **Soares R. de P., Secchi A.** EMSO: A new environment for modelling, simulation and optimisation // *Computer Aided Chemical Engineering*. 2003. V. 14. P. 947–952.
13. **Matsson S. E., Elmquist H., Otter M.** Physical system modeling with Modelica // *Control Engineering Practice*. 1998. V. 6, N 4. P. 501–510.
14. **OpenModelica** — A free open-source environment for system modeling, simulation, and teaching / P. Fritzson, P. Aronsson, A. Pop, H. Lundvall, K. Nystrom, L. Saldamli, D. Broman, A. Sandholm // *IEEE International Symposium on Computer-Aided Control Systems Design*. IEEE. 2006. P. 1588–1595.
15. **Akesson J., Gäfvert M., Tummescheit T. J.** Modelica-an Open Source Platform for Optimization of Modelica Models // *Proceedings of MATHMOD 2009 — 6th Vienna International Conference on Mathematical Modelling*. 2009.

ings of MATHMOD 2009 — 6th Vienna International Conference on Mathematical Modelling. 2009.

16. **Dymola** for multi-engineering modeling and simulation / D. Brüch, H. Elmquist, S. E. Mattsson, H. Olsson // *Proceedings of Modelica*. 2002.
17. **Hřebíček J., Režáč M.** Modelling with Maple and MapleSim // *22nd European Conference on Modelling nad Simulation ECMS 2008 Proceedings*. 2008. P. 60–66.
18. **Li H., Li J.** Modeling and Simulation Software AMESim for Mechanical / Hydraulic System [J] // *Computer Simulation*. 2006. V. 12. 075 p.
19. **A comparison** of metacompilation approaches to implementing Modelica / D. Broman, P. Fritzson, G. Hedin, J. Akesson. Technical Report 1404-1200 Report 97, Dept. of Computer Science, Lund University, Sweden, 2011.
20. **Hairer E., Nørsett S. P., Wanner G.** Solving Ordinary Differential Equations I: Nonstiff Problems: Solving Ordinary Differential Equations. Springer, 2008.
21. **Hairer E., Nørsett S. P., Wanner G.** Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems: Lecture Notes in Economic and Mathematical Systems. Springer, 1996.
22. **Cohen S. D., Hindmarsh A. C.** CVODE, a stiff/nonstiff ODE solver in C // *Computers in physics*. 1996. V. 10, N 2. P. 138–143.
23. **Иванов-Смоленский А. В.** Электрические машины. Т. 1. М.: МЭИ, 2006. 327 с.
24. **Копылов И. П.** Электрические машины. М.: Энергоатомиздат, 1986. 360 с.
25. **Ландау Л. Д., Лифшиц Е. М.** Курс теоретической физики: учеб. пособие для вузов. В 10 т. Т. 1. Механика. М.: Наука, 1988. 214 с.
26. **Vegt O. M.** de- Jet milling from a particle perspective: predicting particle fracture based on mechanical material properties. Groningen, Nederland: Gilderprint Drukkerijen B. V., 2007. 152 p.

N. M. Yakimov, Graduate Student, e-mail: root@kivid.pp.ru,

S. N. Chuvashov, Professor, e-mail: snchuv@mail.ru

"MATI"—Russian State Technological University named after K. E. Tsiolkovsky

Application Software for Composite Mathematical Modelling of Complex Technical Objects

Modern computers make it possible to analyze and model of more complex systems than ever before. However, comparative analysis of current modelling software products shows that they are not perfectly suited for general-purpose modelling of multiphysics systems. We present a new software oriented on general-purpose modelling of complex multiphysics systems, in particular complex technical objects. Our software uses the block-oriented approach to design and visualize of such complex models, and a subset of C++ and XML for description of the block models themselves. An additional software for model analysis and report generation is included. API for user-supplied solvers is simple and consists of one function and one utility class, which makes extension rather simple, compared to other software currently on market. An example application shows significant (up to 4 times) reduction in the model code size in comparison with the traditional approach. The new software is apparently better suited for modelling of complex systems, and application of the widely known programming language makes the learning curve less steep for new researchers.

Keywords: mathematical modelling, application software, block-oriented modelling, multiphysics modelling, modelling of technical objects

References

1. **Rabochie** processy v uzlax turbokompressornogo generatora (kompessor, turbina, e'lektrogenerator) / S. N. Chuvashov, I. B. P'yankov, E. S. Chuvashova, N. M. Yakimov, I. G. Zorina. M.: Pero, 2011. 550 p.
2. **Chuvashova E. S., Chuvashov S. N., Zorina I. G.** Kompleksnaya matematicheskaya model' dlya konceptual'nogo poektirovaniya vysokoskorostnykh letatel'nykh apparatov. *Informacionnye tekhnologii*. 2012. N. 11(195). P. 10–14.
3. **Chuvashova E. S., Chuvashov S. N.** Vybor racional'nykh kharakteristik vysokoskorostnykh letatel'nykh apparatov raznykh masshtabov na osnove kompleksnoj matematicheskoy modeli. *Informacionnye tekhnologii*. 2013. N. 8. P. 12–16.

4. **Popov S. D., Chuvashov S. N.** Razrabotka tekhnologii vyborasushhego kompleksa dlya transportnykh sredstv na vozduшной podushke (TSVP), prednaznachennykh dlya e'kspluatsii na Severe i v Sibiri. *Sb. statej mezhdunarodnoj nauchno-prakticheskoy konferentsii Innovatsionnoe razvitie sovremennoy nauki*. Red. A. A. Sukiasyan. Ufa: RIC BashGU. 2014. V. 3. P. 287–295.

5. **Simulink**, User's Guide, Version 7. The MathWorks Inc, 2009.
6. **VISSIM** User Manual, Version 4.00. PTV AG, Germany, 2004.
7. **Nikoukhah R., Steer S.** SCICOS — a dynamic system builder and simulator. *Computer-Aided Control System Design*, 1996. Proceedings of the 1996 IEEE International Symposium on IEEE, 1996. P. 430–435.
8. **Nikoukhah M. N. R.** Modeling and simulation of differential equations in Scicos. *Modelica Conference*. 2006.

9. **Scicos** FAQ. URL: <http://www.scicos.org/FAQ.html#D2>
10. **EcosimPro**: User Manual. EA International, 2004.
11. **ASCEND**: An object-oriented computer environment for modeling and analysis: The modeling language / P. C. Piela, T. G. Epperly, K. M. Westerberg, A. W. Westerberg. *Computers & chemical engineering*. 1991. V. 15, N 1. P. 53–72.
12. **Soares R. de P., Secchi A.** EMSO: A new environment for modelling, simulation and optimisation. *Computer Aided Chemical Engineering*. 2003. V. 14. P. 947–952.
13. **Matsson S. E., Elmqvist H., Otter M.** Physical system modeling with Modelica. *Control Engineering Practice*. 1998. V. 6, N 4. P. 501–510.
14. **OpenModelica** — A free open-source environment for system modeling, simulation, and teaching / P. Fritzson, P. Aronsson, A. Pop, H. Lundvall, K. Nyström, L. Saldamli, D. Broman, A. Sandholm. *IEEE International Symposium on Computer-Aided Control Systems Design*. IEEE. 2006. P. 1588–1595.
15. **Åkesson J., Gäfvert M., Tummescheit T. J.** Modelica—an Open Source Platform for Optimization of Modelica Models. *Proceedings of MATHMOD 2009 — 6th Vienna International Conference on Mathematical Modelling*. 2009.
16. **Dymola** for multi-engineering modeling and simulation / D. Brück, H. Elmqvist, S. E. Mattsson, H. Olsson. *Proceedings of Modelica*. 2002.
17. **Hřebíček J., Řezáč M.** Modelling with Maple and MapleSim. *22nd European Conference on Modelling nad Simulation ECMS 2008 Proceedings*. 2008. P. 60–66.
18. **LI H., LI J.** Modeling and Simulation Software AMESim for Mechanical / Hydraulic System [J]. *Computer Simulation*. 2006. V. 12. P. 075.
19. **A comparison** of metacompilation approaches to implementing Modelica / D. Broman, P. Fritzson, G. Hedin, J. Åkesson. Technical Report 1404-1200 Report 97, Dept. of Computer Science, Lund University, Sweden, 2011.
20. **Hairer E., Nørsett S. P., Wanner G.** Solving Ordinary Differential Equations I: Nonstiff Problems: Solving Ordinary Differential Equations. Springer, 2008.
21. **Hairer E., Nørsett S. P., Wanner G.** Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems: Lecture Notes in Economic and Mathematical Systems. Springer, 1996.
22. **Cohen S. D., Hindmarsh A. C.** CVODE, a stiff/nonstiff ODE solver in C. *Computers in physics*. 1996. V. 10, N 2. P. 138–143.
23. **Ivanov-Smolenskij A. V.** E'lektricheskie mashiny. V. 1. M.: ME'I, 2006. 327 p.
24. **Kopylov I. P.** E'lektricheskie mashiny. M.: E'nergoatomizdat, 1986. 360 p.
25. **Landau L. D., Lifshic E. M.** Kurs teoreticheskoy fiziki. Uchebnoe posobie dlya VUZov: in 10 v. V. 1. Mexanika. M.: Nauka, 1988. 214 p.
26. **Vegt O. M.** de- Jet milling from a particle perspective: predicting particle fracture based on mechanical material properties. Groningen, Nederland: Gilderprint Drukkerijen B. V., 2007. 152 p.

УДК 004.272.2

А. Л. Стемпковский, академик РАН, директор,
Д. В. Тельпухов, канд. техн. наук, науч. сотр., e-mail: nofrost@inbox.ru,
Р. А. Соловьев, канд. техн. наук, зав. отд.,
А. Н. Соловьев, д-р техн. наук, гл. науч. сотр.,
М. В. Мячиков, студент, инженер-исследователь,
 Институт проблем проектирования в микроэлектронике РАН

Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем

Рассмотрена актуальная задача оценки надежностных характеристик комбинационных схем. Описан аналитический подход к получению надежностных характеристик, который обладает высокой точностью получаемых результатов, однако требует больших вычислительных ресурсов. В противовес аналитическому методу предложен стохастический подход к моделированию возникновения ошибок в комбинационных схемах для получения надежностных характеристик. Описаны различные типы ошибок, модели возникновения ошибок, а также характеристики надежности схемы. Проведено сравнение аналитического метода расчета параметров надежности с предлагаемым методом инжектирования ошибок на примере типовой комбинационной схемы.

Ключевые слова: метод инжектирования ошибок, отказоустойчивость, комбинационная схема, стохастическое моделирование неисправностей

Введение

Развитие современных микроэлектронных средств тесно связано с необходимостью повышения уровня интеграции, быстродействия и надежности. По мере роста степени интеграции и уменьшения технологических размеров возрастает роль повышения надежности и помехоустойчивости проектируемых устройств под воздействием различных источников помех и сбоя: технологических, радиационных,

перекрестных помех, деградации во времени, скачков напряжения питания и др. Особенно остро данная проблема встает в военных, аэрокосмических, медицинских и других областях человеческой деятельности, связанных с высокими рисками пропуска ошибки. В настоящее время складывается ситуация, когда фактор помехоустойчивости в микроэлектронике становится определяющим условием работоспособности и надежности разрабатываемой

электронной аппаратуры, поэтому актуальным является исследование и разработка методов повышения отказоустойчивости микросистемных схем.

Несмотря на достигнутые успехи в разработке кодовых методов защиты, используемых при хранении и передаче данных [1], для обеспечения требуемого уровня отказоустойчивости арифметических и логических схем до сих пор используют архаичные методы кратного резервирования [2]. Таким образом, важной задачей становится разработка эффективных методов повышения характеристик помехозащищенности логических схем. Ключевым аспектом при решении этой проблемы является задача разработки методов оценки эффективности разрабатываемых подходов.

Существующие методы анализа отказоустойчивости комбинационных схем [3] требуют больших вычислительных мощностей и имеют целый ряд методологических недостатков, одним из которых является отсутствие учета реального технологического базиса современных средств проектирования [4]. В известных подходах, как правило, используется абстрактный булевый базис, удовлетворяющий критериям полноты, в то время как в предлагаемом методе используются технологические библиотеки из реального маршрута проектирования, содержащие десятки стандартных ячеек. Таким образом, актуальной является задача разработки методов стохастического моделирования сбоев в процессе верификации схем, выполняемых с учетом реальных технологических базисов проектирования (Modelsim) [5]. Данная статья посвящена разработке методов стохастического моделирования неисправностей логических схем для оценки характеристик отказоустойчивости.

Статья имеет следующую структуру. В разд. I рассмотрены теоретические методы оценки характеристик отказоустойчивости схемы, а также дан пример расчета функции ошибки для схемы прямой импликации и приведены краткие оценки трудоемкости данного подхода. В разд. II представлена классификация ошибок по типу и характеру их возникновения, а также описаны основные принципы метода инжектирования ошибок. Сравнение аналитического метода расчета параметров отказоустойчивости с предлагаемым методом инжектирования ошибок на примере типовой комбинационной схемы приведено в разд. III.

I. Аналитическая оценка вероятности возникновения ошибки на выходе комбинационной схемы

Для аналитической оценки вероятности возникновения ошибки на выходе схемы будем использовать следующий подход. Под ошибкой в конкретном вентиле в схеме будем понимать инверсию его

выходного сигнала, а вектор $\mathbf{e} = (e_1, e_2, \dots, e_M)$ назовем вектором ошибки, где

$$e_i = \begin{cases} 1, & \text{если на } i\text{-м элементе произошла ошибка;} \\ 0, & \text{иначе.} \end{cases}$$

Цель данного метода — получение полинома ошибки EOF(p), который определяет вероятность возникновения такой комбинации входных сигналов и вектора ошибки, которая приводит к искажению результата работы функции. Аргументом этого полинома является параметр p , обозначающий вероятность ошибки в одном вентиле схемы. Таким образом, полином ошибки является характеристикой, отражающей способность схемы противостоять случайным сбоям ее элементов.

Для примера в качестве исследуемой схемы выберем схему прямой импликации, построенную в базисе И-НЕ (рис. 1).

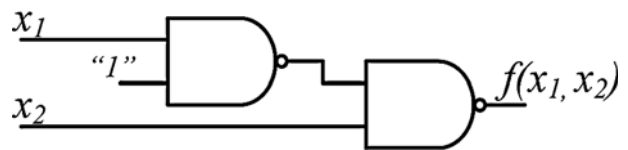


Рис. 1. Структурная схема прямой импликации в базисе И-НЕ

Эта схема состоит из двух элементов NAND, соединенных последовательно. Обозначим результат эталонной схемы как $f(x_1, x_2)$, а результат схемы, подверженной ошибкам, как $f^*(x_1, x_2, e_1, e_2)$. Схема, подверженная ошибкам, изображена на рис. 2.

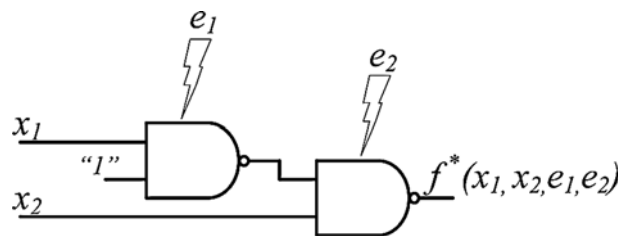


Рис. 2. Структурная схема прямой импликации в базисе И-НЕ, подверженная внешнему воздействию

Формально в схеме появилось два дополнительных входа e_1 и e_2 , которые инвертируют выходные значения на соответствующих логических вентилях.

Введем характеристическую функцию [6] множества пар векторов входного сигнала \mathbf{X} и ошибки \mathbf{e} :

$$E(\mathbf{X}, \mathbf{e}) = \begin{cases} 1, & \text{если набор } (\mathbf{X}, \mathbf{e}) \text{ приводит к ошибке;} \\ 0, & \text{иначе.} \end{cases}$$

Построим полную таблицу истинности (табл. 1) по всем возможным комбинациям входов и векторов ошибки (\mathbf{X}, \mathbf{e}) .

В последнем столбце указаны вероятности появления вектора ошибки при условии того, что вероятность возникновения ошибки на каждом вентиле

Таблица 1

Таблица истинности по всем (\mathbf{X}, \mathbf{e}) для схемы прямой импликации

№	x_1	x_2	e_1	e_2	$f(x_1, x_2)$	$f^*(x_1, x_2, e_1, e_2)$	$E(\mathbf{X}, \mathbf{e})$	Вероятность возникновения вектора ошибки
0	0	0	0	0	1	1	0	$(1-p)^2$
1	0	0	0	1	1	0	1	$(1-p)p$
2	0	0	1	0	1	1	0	$p(1-p)$
3	0	0	1	1	1	0	1	p^2
4	0	1	0	0	0	0	0	$(1-p)^2$
5	0	1	0	1	0	1	1	$(1-p)p$
6	0	1	1	0	0	1	1	$p(1-p)$
7	0	1	1	1	0	0	0	p^2
8	1	0	0	0	1	1	0	$(1-p)^2$
9	1	0	0	1	1	0	1	$(1-p)p$
10	1	0	1	0	1	1	0	$p(1-p)$
11	1	0	1	1	1	0	1	p^2
12	1	1	0	0	1	1	0	$(1-p)^2$
13	1	1	0	1	1	0	1	$(1-p)p$
14	1	1	1	0	1	0	1	$p(1-p)$
15	1	1	1	1	1	1	0	p^2

равна p . Например, вероятность появления вектора ошибки $\mathbf{e} = (e_1, e_2) = (0, 1)$ равна $(1-p)p$, так как отсутствие ошибки в вентиле, соответственно, равно $(1-p)$.

Нас интересуют строки, в которых значение функции не совпадает с эталонным, а иными словами — строки, в которых характеристическая функция $E(\mathbf{X}, \mathbf{e})$ равна единице. Обратившись к строчке под номером 5 в таблице и учитывая тот факт, что вероятность любой комбинации двух входов равна $\frac{1}{4}$, получим вероятность наступления этого события: $\frac{1}{4}(1-p)p$. Если для каждой строчки, в которой $E(\mathbf{X}, \mathbf{e}) = 1$ найти вероятность ее возникновения, то, просуммировав все эти выражения, мы получим полином ошибки, который характеризует вероятность несовпадения результата работы схемы с эталонным при вероятности ошибки на вентиле, равной p . Для рассмотренного примера полином будет выглядеть следующим образом:

$$EOF(p) = (1-p)p + p^2 + (1-p)p + p(1-p) + (1-p)p + p^2 + (1-p)p + p(1-p) = 6p - 4p^2.$$

Теперь выведем формулу для расчета полинома ошибки в общем виде для произвольной логической схемы. Учитывая, что вероятность появления на входе конкретного вектора входных сигналов \mathbf{X} длины N (в предположении равномерности всех таких наборов) равна $\frac{1}{2^N}$, а вероятность возникнове-

ния вектора ошибки \mathbf{e} длины M и веса $|\mathbf{e}|$ равна $p^{|\mathbf{e}|}(1-p)^{M-|\mathbf{e}|}$, получаем вероятность ошибки на выходе схемы (вес вектора равен числу его ненулевых элементов):

$$EOF(p) = \frac{1}{2^N} \sum_{\mathbf{X}, \mathbf{e}} E(\mathbf{X}, \mathbf{e}) p^{|\mathbf{e}|} (1-p)^{M-|\mathbf{e}|}, \quad (1)$$

где суммирование ведется по всем возможным комбинациям (\mathbf{X}, \mathbf{e}) .

Вероятность правильного значения, соответственно, может быть найдена по формуле

$$COF(p) = 1 - EOF(p). \quad (2)$$

Рассмотренный пример отчетливо иллюстрирует основной недостаток этого подхода, который связан с экспоненциальной скоростью роста вычислительной сложности от числа элементов схемы. Из-за этого недостатка данный метод применим только на участках схемы с небольшим числом элементов, в то время как типовые комбинационные схемы, например 16-битный умножитель или сложная логическая управляющая схема, могут состоять из сотен и тысяч логических вентилях, где теоретический метод будет неприменим.

Одним из вариантов решения обозначенной проблемы могут являться стохастические (симуляционные) методы, основанные на современных системах моделирования. К данному типу методов можно отнести предлагаемый метод инъектирования ошибок, который будет подробно рассмотрен в следующем разделе.

II. Метод инъектирования ошибок

Суть метода заключается в том, что в процессе моделирования исследуемой логической схемы в один или несколько узлов схемы вносится ошибка. Необходимо определить: произошло ли искажение результата работы схемы и была ли ошибка обнаружена.

Предлагаемый подход инъектирования ошибок при логическом моделировании подразумевает модификацию схемы таким образом, чтобы появилась возможность вносить ошибки в любой узел схемы. Для этого после каждого вентиля в схему необходимо добавить дополнительный элемент, называемый инжектором. Процесс модификации схемы на примере логической функции XOR в базисе И-ИЛИ-НЕ представлен на рис. 3.

Первый вход инжектора соединяется с выходом вентиля, выход инжектора подается в соответствии с начальной коммутацией схемы. Второй входной сигнал инжектора определяется как входной сигнал основной схемы и интерпретируется как дополнительный вход. Логическая единица на любом дополнительном входе означает ошибку в соответствующем ей вентилю.

Инжектор выполняет бинарную логическую операцию, формируемую в зависимости от типа моделируемой ошибки.

II.1. Типы моделируемых ошибок

Чаще всего в литературе, связанной с помехоустойчивостью логических схем, рассматриваются только ошибки, связанные с инверсией сигнала. Однако часто на практике приходится встречаться с ошибками, обусловленными технологическими несовершенствами микроэлектронных изделий, таких как замыкания. В этой работе мы будем рассматривать три основных типа ошибок:

- замыкание на питание (stuck at VCC);
- замыкание на землю (stuck on ground);
- инверсия сигнала.

Структура модифицированной схемы в зависимости от типа ошибок существенно изменяться не будет, меняются только двухвходовые логические элементы, эмулирующие ошибку в вентиле. Типы элементов с таблицами истинности для каждого типа ошибки представлены в табл. 2.

Метод статистического внесения сбоя заключается в использовании метода Монте-Карло [7], что подразумевает проведение многократных экспериментов с различными входными данными и последующей оценкой необходимых параметров отказоустойчивости.

II.2. Модели возникновения ошибок

В общем виде схема для оценки помехоустойчивости представлена на рис. 4.

Пусть входы для моделирования ошибок формируют вектор ошибки:

$$\mathbf{e} = (e_1, e_2, \dots, e_n),$$

где e_i — i -й вход. При $e_i = 1$ в схему внедряется ошибка в i -й вентиль. При $e_i = 0$ — ошибки нет.

Шум окружающей среды (environmental noise). При реализации этой стратегии внесения ошибок будем рассматривать ситуацию, когда каждый вентиль имеет некоторую вероятность сбоя. Для удобства эта вероятность фиксируется для всех вентилях схемы, а ошибки происходят независимо.

Таким образом, введем некоторый вещественный параметр $\gamma \in [0, 1]$, который будет являться характеристикой шума окружающей среды. Тогда элемент e_i при каждой подаче входных воздействий с вероятностью γ может обратиться в единицу. При таком подходе количество ошибок в схеме, при одной подаче данных, лежит в диапазоне от нуля до общего числа элементов схемы n . Случаи с нулевым вектором ошибки при подсчетах характеристик помехоустойчивости схемы не рассматриваются.

Кратные ошибки. Фиксируется параметр $k \in \{0, 1, 2, \dots, n\}$ — кратность ошибки. Вектор \mathbf{e} при каждой подаче входных воздействий генерируется случай-

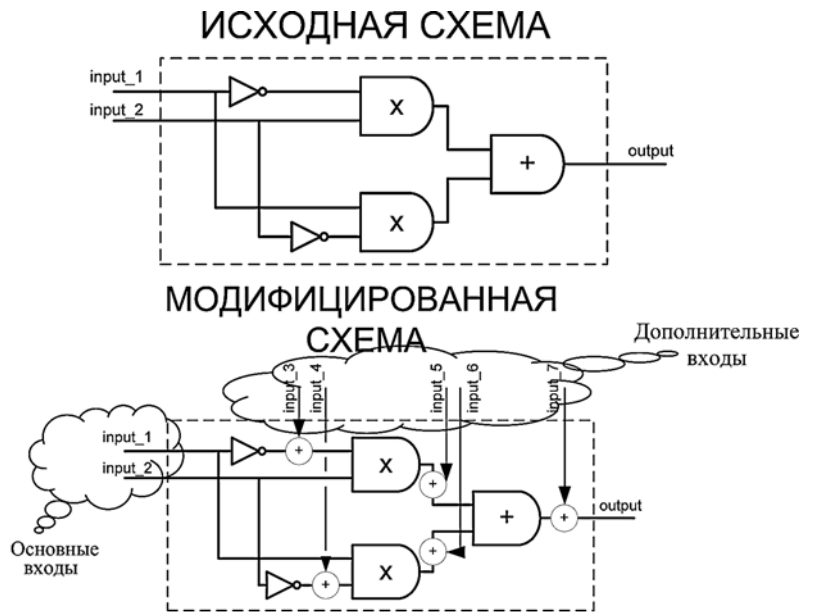


Рис. 3. Метод инжектирования ошибки

Таблица 2

Инжекторы для различных типов ошибок

Тип ошибки	Таблица истинности для инжектора			Название булевой функции инжектора
	from_gate	error	out	
Замыкание на питание (stuck at VCC)	0	0	0	OR, логическое ИЛИ
	0	1	1	
	1	0	1	
	1	1	1	
Замыкание на землю (stuck on ground)	0	0	0	Инверсия прямой импликации ($\overline{x \rightarrow y}$), больше, детектор 4
	0	1	0	
	1	0	1	
	1	1	0	
Инверсия сигнала	0	0	0	XOR, сумма по модулю 2, исключающее ИЛИ
	0	1	1	
	1	0	1	
	1	1	0	

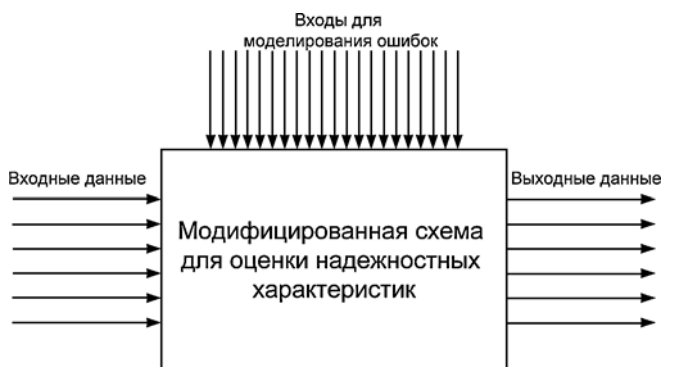


Рис. 4. Структурная схема метода инжектирования ошибки

ным образом так, чтобы вес его всегда был равен k . Пример векторов для $k = 2$:

- (0,1,0,0,1,0,0)
- (1,0,0,0,0,0,1)
- (0,0,0,1,1,0,0)
- (0,0,1,0,0,1,0).

Стресс-тесты для определенных вентилях. Кроме нахождения обобщенных характеристик отказоустойчивости схемы, метод инжектирования ошибок может быть полезен для выявления наиболее уязвимых участков схемы с точки зрения влияния ошибки в конкретном вентиле на общую работоспособность. Эмпирически было установлено существенное различие в уязвимости для разных участков типовых схем. Этот факт приводит к идее разработки методов селективной избыточности комбинационных схем. Эффективным методом определения степени уязвимости участка схемы может служить метод инжектирования ошибок с использованием стратегии стресс-теста:

— фиксируется элемент e_i , который на всем протяжении тестирования будет равен единице, остальные элементы вектора ϵ обращены в ноль;

— определяется влияние конкретного вентиля (группы вентилях) на корректность работы всей схемы в условиях помех.

Селективное добавление избыточности для наиболее уязвимых участков схемы позволит достичь разумного компромисса в надежности функционирования при незначительном увеличении площади микросхемы.

II.3. Характеристики помехозащищенности схемы

Для оценки архитектурной устойчивости к сбоям и отказам, а также для оценки эффективности методов исправления ошибок в процессе моделирования необходимо определять следующие параметры:

- общее число экспериментов;
- количество корректных значений выхода схемы;
- общее число внедренных ошибок.

После проведения серии экспериментов в количестве, достаточном для получения корректной статистической выборки, необходимо вычислить характеристики, с помощью которых можно эффективно сравнивать между собой различные архитектуры комбинационных схем с точки зрения их помехозащищенности.

Для оценки отказоустойчивости схем с точки зрения исправления ошибок будем рассматривать два параметра. Первый параметр характеризует архитектурную устойчивость к ошибкам и определяется как отношение числа некорректных результатов работы схемы к общему числу инжектированных ошибок:

$$T_o = \frac{I_{cr}}{T_{er}}, \quad (3)$$

где I_{cr} — число экспериментов с несовпадением результата с эталонным; T_{er} — общее число внедренных ошибок. По сути, этот параметр определяет вероятность того, что внедренная ошибка в схему приведет к искажению работы всей схемы.

Второй параметр определяет общую устойчивость схемы к ошибкам и вычисляется как отношение числа корректных результатов работы схемы к общему числу тестов:

$$A = \frac{C_r}{T_r}. \quad (4)$$

В случае комбинационных схем, обнаруживающих ошибки, мы располагаем не только эталонным значением выхода схемы, но и эталонным значением флага ошибки на заданных входных данных. Таким образом, в процессе моделирования необходимо подсчитывать число появлений для событий, описанных в табл. 3.

Таблица 3

Характеристики помехоустойчивости для схем с обнаружением ошибок

Истинность результата (0 — результат верный; 1 — результат неверный)	Флаг ошибки	Событие
0	0	Ошибка не "проявилась"
0	1	"Ложная тревога"
1	0	Ошибка пропущена
1	1	Ошибка обнаружена

Следует отметить, что при каждой подаче данных в процессе моделирования в схему вносится, по меньшей мере, одна ошибка. В этом контексте первую строку табл. 3 следует понимать, как некоторую способность архитектуры маскировать ошибки, т. е. ошибка, возникшая на одном из вентилях схемы, не доходит до выходов вследствие особенностей ее архитектуры.

Эти четыре варианта составляют полную группу событий, следовательно, характеристиками помехоустойчивости схемы будут являться вероятности наступления каждого из событий.

III. Оценка помехоустойчивости библиотечных бенчмарк-схем

Описанный метод был выполнен в виде программы, реализующей модифицированную схему в формате Verilog HDL из вентильного представления синтезированной схемы в САПР Synopsys Design Compiler. Кроме того, программа реализует модуль Testbench для оценки необходимых параметров. Таким образом, еще одним этапом в предлагаемом маршруте является моделирование модифицированной схемы в среде моделирования ModelSim.

Чтобы оценить эффективность метода инжектирования ошибки с точки зрения точности получаемых оценок помехоустойчивости, было проведено

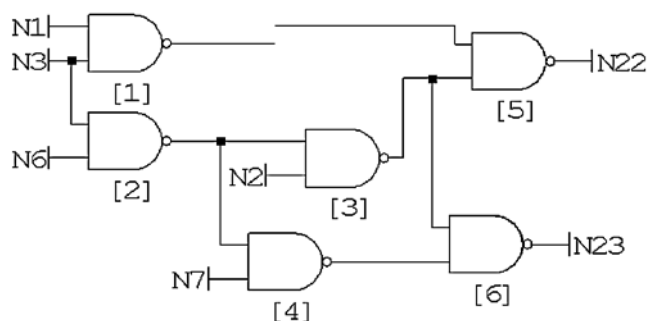


Рис. 5. Бенчмарк схема ISCAS85 C17

моделирование для оценки параметра устойчивости к ошибкам (4) стандартной схемы ISCAS85 C17 [8], приведенной на рис. 5, после синтеза в рамках библиотеки Nangate 45 nm [9]. Параметр число прогонов моделирования установлен на значение 10 000.

Проводилось сравнение с теоретической оценкой, полученной с помощью формулы (2) в диапазоне изменения характеристики шума окружающей среды от нуля до 50%. Полученные результаты приведены на рис. 6 (см. вторую сторону обложки).

Оценивая полученные результаты, можно сделать вывод о том, что для рассмотренных схем теория хорошо сходится с практикой и есть основания полагать, что метод окажется столь же успешным и для схем больших размерностей. Кроме того, варьируя число прогонов моделирования, можно достигнуть приемлемых значений точности результатов при сохранении требуемой производительности метода.

Заключение

В работе был рассмотрен стохастический метод инжектирования ошибок для оценки параметров помехоустойчивости комбинационных микроэлек-

тронных устройств. Были рассмотрены разные модели возникновения ошибок, типы ошибок, а также характеристики помехозащищенности. Было реализовано программное средство, а также разработан маршрут для оценки помехоустойчивости схем методом инжектирования ошибок. С его помощью были получены оценки помехозащищенности для типовой комбинационной схемы и проведено сравнение с аналитическим методом расчета параметров отказоустойчивости. Было показано, что метод инжектирования ошибок вкупе с современными средствами моделирования позволяет осуществлять оценку характеристик помехоустойчивости для достаточно больших логических схем, в то время как теоретические методы существенно ограничены числом элементов схемы в связи с экспоненциальным ростом вычислительной сложности.

Список литературы

1. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. М.: Техносфера, 2006. 320 с.
2. El-Maleha A. H., Oughalia F. C. A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits // Microelectronics Reliability. 2014. Vol. 54, N 1. P. 316–326.
3. Bhaduri D. Tools and Techniques for Evaluating Reliability Trade-Offs for Nano-Architectures. Virginia: Virginia Polytechnic Institute and State University, 2004.
4. Han J., Gao J., Qi Y., Jonker P., Fortes J. Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics // IEEE Design & Test of Computers. 2005. Vol. 22, N. 4. P. 328–339.
5. Ziade H., Ayoubi R., and Velazco R. A Survey on Fault Injection Techniques // The International Arab Journal of Information Technology. July 2004. Vol. 1, N. 2.
6. Folland G. B. Real Analysis: Modern Techniques and Their Applications (Second ed.). John Wiley & Sons, Inc. 1999.
7. Бахвалов Н. С., Воеводин В. В. Современные проблемы вычислительной математики и математического моделирования. Том 1. Вычислительная математика. М.: Наука, 2005. 343 с.
8. Hansen M., Yalcin H., Hayes J. P. Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering // IEEE Design and Test. 1999. Vol. 16, N. 3. P. 72–80.
9. http://www.nangate.com/?page_id=2325

A. L. Stempkovskiy, Academician, Director, D. V. Telpukhov, Ph. D., Researcher,
R. A. Solovyev, Ph. D., Head of the Department, A. N. Solovyev, Ph. D., Chief Scientist
M. V. Myachikov, Student, Research Engineer
Institute for Design Problem in Microelectronics

Fault Simulation Technique for Logic Circuits Reliability Characteristics Evaluation

The article considers the actual problem of estimating fault tolerance characteristics for combinational circuits. Analytical approach for obtaining fault tolerance characteristics is described. High accuracy of the method that is achieved by using large computational resources is demonstrated on particular examples. In contrast to the analytical method, stochastic approach for error modeling in the combinational circuits for obtaining fault tolerance characteristics is proposed. Different types of faults, fault models, and fault tolerance characteristics of the circuit are described. Software tool was implemented and evaluation route for the fault tolerance of combinational circuits via fault injection technique is developed. A comparison of the analytical method for calculating the fault tolerance parameters and the proposed method of injecting errors is carried out on typical benchmark combinational circuit. Comparison showed a good convergence of stochastic method with analytical approach.

Keywords: fault injection method, fault tolerance, combinational circuit, stochastic modeling of faults

References

1. **Morelos-Saragosa R.** *Iskusstvo pomehoustojchivogo kodirovaniya. Metody, algoritmy, primenenie.* M.: Tehnosfera, 2005. 320 p.
2. **El-Maleha A. H., Oughalia F. C.** A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits. *Microelectronics Reliability.* 2014. Vol. 54, N 1. P. 316–326.
3. **Bhaduri D.** *Tools and Techniques for Evaluating Reliability Trade-Offs for Nano-Architectures.* Virginia: Virginia Polytechnic Institute and State University, 2004.
4. **Han J., Gao J., Qi Y., Jonker P., Fortes J.** Toward Hardware-Redundant, Fault-Tolerant Logic for Nanoelectronics. *IEEE Design & Test of Computers.* 2005. Vol. 22, N 4. P. 328–339.
5. **Ziade H., Ayoubi R., Velazco R.** A Survey on Fault Injection Techniques. *The International Arab Journal of Information Technology.* July 2004. Vol. 1. N 2.
6. **Folland G. B.** *Real Analysis: Modern Techniques and Their Applications* (Second ed.). John. Wiley & Sons, Inc., 1999.
7. **Bahvalov N. S., Voevodin V. V.** *Sovremennye problemy vychislitel'noj matematiki I matematicheskogo modelirovaniya. Vol. 1. Vychislitel'naja matematika.* M.: Nauka, 2005. 343 p.
8. **Hansen M., Yalcin H., Hayes J. P.** Unveiling the ISCDAS-85 Benchmarks: A Case Study in Reverse Engineering *IEEE Design and Test.* 1999. Vol. 16. N 3. P. 72–80.
9. http://www.nangate.com/?page_id=2325

УДК 004.94

А. А. Усков, д-р техн. наук, проф., e-mail: andrey@uskov.net, А. Г. Жукова,
Российский университет кооперации

Матричное представление иерархических визуальных моделей для CASE-средств

Рассмотрено предложенное авторами матричное представление структуры иерархического группирования объектов, отличающееся использованием аппарата многомерных матриц, что позволяет достичь компактного математического описания. Матричное представление может использоваться для описания структуры иерархических визуальных моделей (методологии IDEF, UML, ARIS и др.).

На основе матричного представления предложен подход к оценке сложности восприятия человеком иерархической визуальной модели, в этом подходе используются принципы функционирования оперативной памяти человека.

Результаты работы могут найти применение при разработке CASE-средств.

Ключевые слова: CASE-средства, иерархическое группирование объектов, многомерная матрица, коэффициент сложности восприятия

Введение

Основой современных CASE-средств являются визуальные модели (методологии IDEF, UML, ARIS и др.) [1–4]. Многие визуальные модели строят по иерархическому принципу, когда диаграммы нижнего уровня детализуют элементы (блоки) диаграмм верхнего уровня. При этом иерархия моделей может быть представлена древовидной структурой [3].

Ниже рассмотрено предложенное авторами матричное представление иерархического группирования объектов (ИГО), в компактной форме описывающее связь между диаграммами различного уровня и числом объектов на нижнем уровне иерархии.

На основе матричного представления ИГО рассмотрен подход к оценке сложности восприятия иерархических визуальных моделей (SADT (IDEF0), DFD, Use Case Diagram и др.) на основе разработанного представления.

Результаты статьи могут найти применение при разработке CASE-средств.

1. Матричное представление иерархического группирования

Рассмотрим иерархическое группирование объектов в классификационные группировки, представленное на рис. 1.

Имеется N объектов, на первом уровне группирования все объекты разделяются на n_1 группировок с номерами $i_1 = 1, 2, \dots, n_1$.

На втором уровне группирования каждая из группировок первого уровня делится на ряд подгруппировок. Так, 1-я группировка ($i_1 = 1$) делится на $n_{2,1}$ подгруппировок с номерами $i_2 = 1, 2, \dots, n_{2,1}$.

Описанное разделение на подгруппировки проводится l раз (l уровней группирования). Подгруппировки нижнего уровня, путь к которым по дереву группирования от корня определяется совокупностью группировок с номерами i_1, i_2, \dots, i_l содержат объекты в количестве m_{i_1, i_2, \dots, i_l} (см. рис. 1).

Рассмотрим пример наиболее распространенного двухуровневого группирования ($l = 2$) (рис. 2).

Число группировок на уровне 1 — n_1 . На уровне 2 — $n_{2,1}, n_{2,2}, \dots, n_{2,n_1}$.

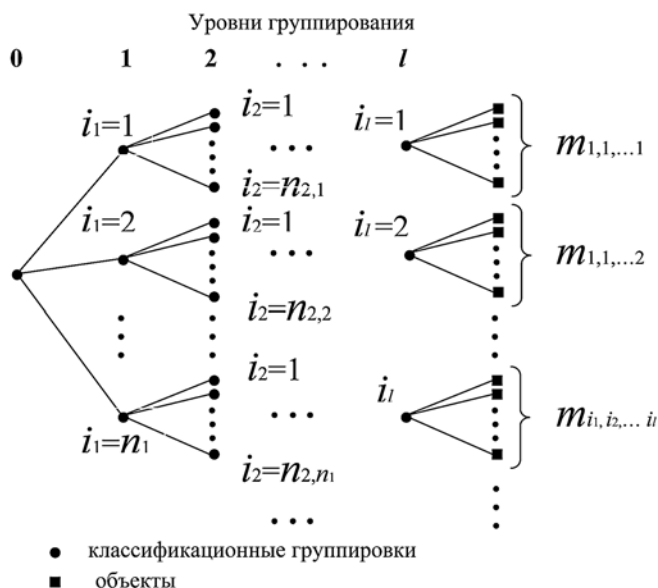


Рис. 1. Общая схема иерархического группирования

Число классифицируемых объектов определяется формулой

$$N = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_{2i_1}} m_{i_1, i_2}. \quad (1)$$

Под структурой иерархического группирования объектов будем понимать совокупность данных о числе классифицирующих группировок на каждом уровне, числе подгруппировок, относящихся к данной группировке, и числе объектов в каждой подгруппировке нижнего уровня.

Отметим, что максимальный размер группировки на уровне 2

$$n_2 = \max(n_{2,1}, n_{2,2}, \dots, n_{2n_1}). \quad (2)$$

Запишем структуру ИГО в виде матрицы размерности $n_1 \times n_2$:

$$M = \begin{matrix} [m_{i_1, i_2}] \\ i_1=1, 2, \dots, n_1; i_2=1, 2, \dots, n_2 \end{matrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,n_2} \\ m_{2,1} & m_{2,2} & \dots & m_{2,n_2} \\ \dots & \dots & \dots & \dots \\ m_{n_1,1} & m_{n_1,2} & \dots & m_{n_1,n_2} \end{bmatrix}, \quad (3)$$

где m_{i_1, i_2} — число элементов в группировке i_1 и подгруппировке i_2 .

Число элементов в группировке i_1 определяется формулой

$$N_{i_1} = \sum_{i_2=1}^{n_2} m_{i_1, i_2}. \quad (4)$$

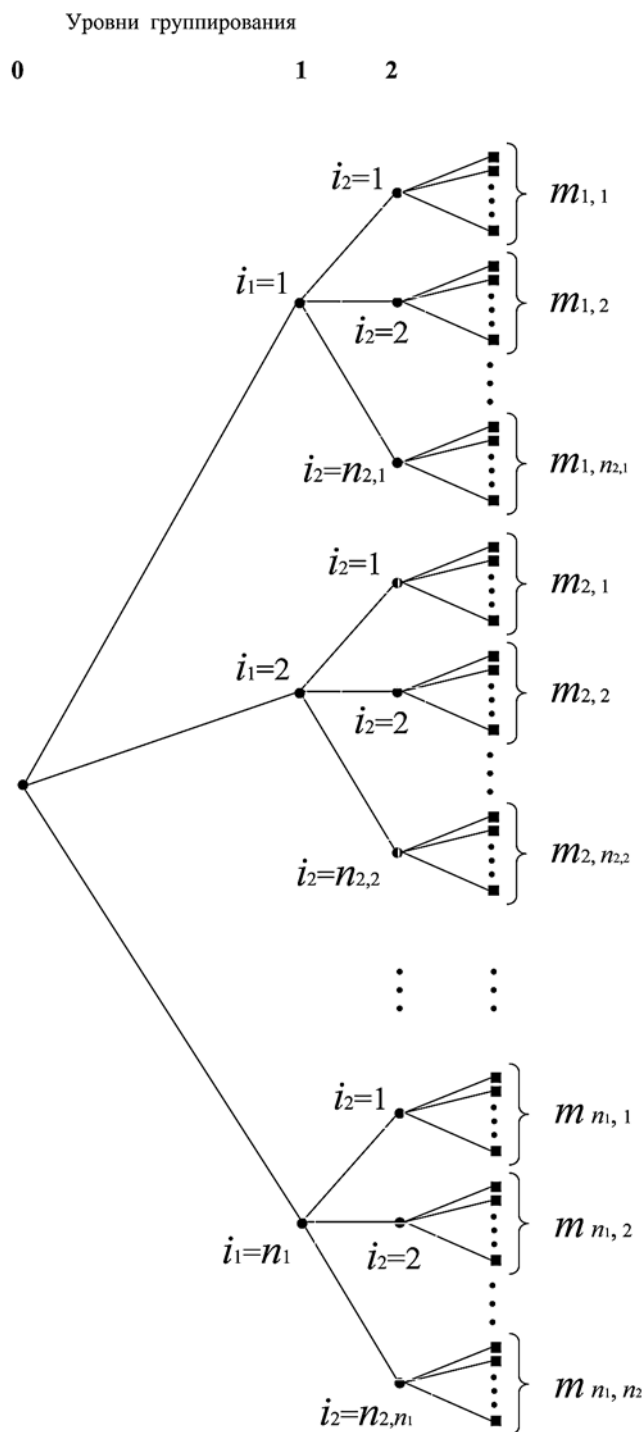


Рис. 2. Схема двухуровневого иерархического группирования

Общее число объектов в соответствии с формулой (1)

$$N = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} m_{i_1, i_2} = \text{sum sum}(M), \quad (5)$$

где $\text{sum sum}(M)$ — сумма элементов матрицы M .

Пример 1. Рассмотрим двухуровневое иерархическое группирование объектов, показанное на рис. 3.

Уровни
группирования

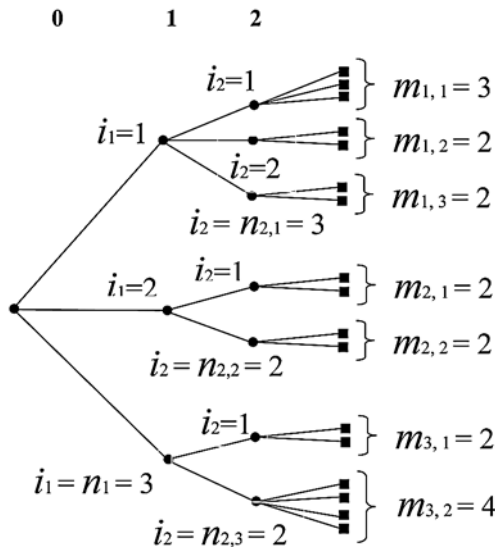


Рис. 3. Пример двухуровневого ИГО

В рассматриваемом случае число группировок уровня 1 $n_1 = 3$, а максимальный размер группировки на уровне 2 согласно формуле (2) $n_2 = \max(n_{2,1}, n_{2,2}, n_{2,3}) = \max(3, 2, 2) = 3$.

Матрица структуры ИГО согласно формуле (3) имеет вид:

$$M = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 2 & 0 \\ 2 & 4 & 0 \end{bmatrix}$$

Число объектов в каждой из классификационных группировок, согласно формуле (4): $N_1 = 7$, $N_2 = 4$, $N_3 = 6$; в соответствии с формулой (5) общее число объектов $N = 17$, что соответствует числу, приведенному на рис. 3.

Если число уровней группирования больше 2 ($l > 2$), то матричное представление структуры ИГО может быть построено аналогично рассмотренному, но матрица M будет l -мерной [5].

2. Оценка сложности восприятия иерархических визуальных моделей

Физиологические исследования обнаруживают два основных вида памяти человека: кратковременная и долговременная [6]. Кратковременная память позволяет вспомнить что-либо через промежуток времени от нескольких секунд до минуты без повторения. Емкость кратковременной памяти составляет 7 ± 2 объекта. Именно такое число объектов хорошо поддается счету, классификации и длительному запоминанию за счет эффективного переноса в долговременную память из кратковременной памяти. Более детальные исследования показывают связь между числом информационных единиц, силой ингибиторных связей между нейронами, веро-

ятностью ошибок и энергетическими затратами на запоминание. Связь между числом информационных единиц и энергетическими затратами на запоминание в относительных единицах приведена в таблице [7].

Связь между числом информационных единиц n и энергетическими затратами на запоминание Q в относительных единицах

n	3	7	10	13	15
Q	1	15	50	200	400

Достаточно точной аппроксимацией зависимости $Q(n)$ является функция

$$Q(n) = \exp(0,4n) - 1. \quad (6)$$

Для дальнейшего анализа введем ряд допущений: 1) сложность восприятия иерархической визуальной модели пропорциональна энергетическим затратам на ее запоминание; 2) человек изучает модель достаточно медленно, последовательно начиная с диаграммы верхнего уровня, затем переходя к диаграммам все более низкого уровня, информация о содержании диаграммы успевает перейти из кратковременной в долговременную память за время рассмотрения одной диаграммы.

С учетом принятых допущений и формулы (6) введем в рассмотрение для визуальных моделей коэффициент сложности восприятия (КСВ), определяемый выражениями

$$Q_s = \sum_{i=1}^K Q_i; \quad Q_i = \exp(0,4n_i) - 1, \quad (7)$$

где n_i — число блоков на i -й диаграмме; K — общее число диаграмм декомпозиции для всех уровней модели; Q_i — КСВ для i -й диаграммы.

Рассмотрим формулу для расчета КСВ иерархической модели.

Для уровня 1 с учетом формулы (7)

$$Q_1 = Q(n_1) = Q(\text{length}(M)), \quad (8)$$

где $\text{length}(M)$ — число строк матрицы M .

На уровне 2 с учетом (7)

$$Q_2 = \sum_{i_1=1}^{n_1} Q(n_{2,i_1}) = \text{sum}(Q(zn(M))), \quad (9)$$

где $zn(M, i_2)$ — функция, возвращающая вектор с числом ненулевых элементов в строках матрицы M (точка после матрицы M означает покомпонентное вычисление функции).

Для самого нижнего уровня (уровня объектов) с учетом (5) и (7) можно записать:

$$Q_3 = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} Q(m_{i_1, i_2}) = \text{sum sum}(Q(M)). \quad (10)$$

Общий КСВ модели определяется суммой Q_1 (см. (8)), Q_2 (см. (9)), Q_3 (см. (10)):

$$Q_s = Q(n_1) + \text{sum}(Q(zn(M))) + \text{sum}(Q(M)). \quad (11)$$

Пример 2. Предположим, имеется двухуровневая модель с матрицей структуры ИГО из примера 1. Определим для нее КСВ согласно формуле (11), используя систему MATLAB 6.5. [8].

Программный код представлен ниже.

```
function [F] = F(x)
% Расчет КСВ
M = [3 2 2; 2 2 0; 2 4 0] % Задается матрица структуры ИГО
Q_s = Q(length(M)) + sum(Q(sum(spones(M')))) + sum(Q(M))
%-----
function [Q] = Q(n) % Расчет Q(n)
Q = exp(0.4*n) - 1;
```

В результате выполнения расчетов получим $Q_s = 19,49$.

Заключение

Предложенное в статье матричное представление структуры иерархического группирования объектов, а также использующий его подход к оценке сложности восприятия человеком визуальных иерархических моделей могут найти применение при

разработке CASE-средств. В частности, предполагается разработать алгоритмы и реализующие их программные приложения, которые позволят в процессе взаимодействия с пользователем в диалоговом режиме строить оптимальные, с точки зрения сложности восприятия, иерархические визуальные модели на основе методологий IDEF, UML, ARIS и др., что, в свою очередь, приведет к улучшению технологии разработки программного обеспечения.

Список литературы

1. **Stahlknecht P., Hasenkamp U.** Einführung in die Wirtschaftsinformatik. Berlin—Heidelberg—New York: Springer—Verlag, 2012. 327 s.
2. **Sommerville I.** Software Engineering (9th Ed.). Boston—Columbus—Indianapolis—New York: Addison-Wesley, 2011. 790 p.
3. **Вендров А. М.** Проектирование программного обеспечения экономических информационных систем. 2-е изд., перераб. и доп. М.: Финансы и статистика, 2006. 544 с.
4. **Черемных С. В., Семенов И. О., Ручкин В. С.** Моделирование и анализ систем. IDEF-технологии: практикум. М.: Финансы и статистика, 2006. 192 с.
5. **Соколов Н. П.** Введение в теорию многомерных матриц. Киев: Наукова думка, 1972. 177 с.
6. **Саймон Г.** Науки об искусственном. М.: Едиториал УРСС, 2004. 144 с.
7. **Bick C., Rabinovich M. I.** Dynamical Origin of the Effective Storage Capacity in the Brain's Working Memory // *Physical Review Letters*. 2009. V. 103. 218101.
8. **Дьяконов В. П.** MATLAB 6.5 SP1/7.0 + Simulink 5/6. Основы применения. М.: Солон-Пресс, 2005. 806 с.

A. A. Uskov, Professor, e-mail: prof.uskov@gmail.com, A. G. Zhukova,
Russian University of Cooperation

Matrix Representation Hierarchical Visual Models for Case-Tools

The article discusses the proposed matrix representation of the hierarchical grouping of objects, using different apparatus multidimensional matrices, which allows for a compact mathematical description. Matrix representation used to describe the structure of hierarchical visual patterns (methodology IDEF, UML, ARIS etc.).

On the basis of the matrix representation of the approach to assessing the complexity of human perception of hierarchical visual model using principles of human memory.

Our results can be applied in the development of CASE-tools.

Keywords: CASE-tools, hierarchical grouping of objects, multidimensional matrix, factor of perception

References

1. **Stahlknecht P., Hasenkamp U.** Einführung in die Wirtschaftsinformatik. Berlin—Heidelberg—New York: Springer—Verlag, 2012. 327 s.
2. **Sommerville I.** Software Engineering (9th Ed.). Boston—Columbus—Indianapolis—New York: Addison-Wesley, 2011. 790 p.
3. **Vendroy A. M.** Projektirovanie programmnogo obespechenija je-konomicheskikh informacionnyh sistem. 2-e izd., pererab. i dop. M.: Finansy i statistika, 2006. 544 p.
4. **Cheremnyh S. V., Semenov I. O., Ruchkin V. S.** Modelirovanie i analiz sistem. IDEF-tehnologii: praktikum. M.: Finansy i statistika, 2006. 192 p.
5. **Sokolov N. P.** Vvedenie v teoriju mnogomernyh matricy. Kiev: Naukova dumka, 1972. 177 p.
6. **Sajmon G.** Nauki ob iskusstvennom. M.: Editorial URSS, 2004. 144 p.
7. **Bick C., Rabinovich M. I.** Dynamical Origin of the Effective Storage Capacity in the Brain's Working Memory. *Physical Review Letters*. 2009. V. 103. 218101.
8. **D'jakonov V. P.** MATLAB 6,5 SP1/7.0 + Simulink 5/6. Osnovy primenenija. M.: Solon-Press, 2005. 806 p.

УДК 004.9

И. Г. Игнатова, д-р техн. наук, проф., проректор по учебной работе, e-mail: igi@miee.ru,
Е. О. Берёза, нач. отдела,
Национальный исследовательский университет "МИЭТ"

Модель интеграции гетерогенных информационных ресурсов с гибким управлением связями

Рассмотрена модель интеграции и хранения гетерогенных информационных ресурсов в виде объектов, их атрибутов и связей между ними, которая обеспечивает принципиальное единообразие хранения всех данных, а также гибкость в управлении системой ограничений, накладываемых на значения атрибутов объектов и связей между ними. Использование данной модели показано на примере соблюдения требования федерального государственного образовательного стандарта 3-го поколения по обеспечению основных образовательных программ учебно-методической документацией и содержательными материалами. Приведена классификация атрибутов объектов модели по родам и видам в разрезе определенных признаков. Описаны два способа гибкого управления связями между гетерогенными информационными ресурсами: с использованием специального объекта типа "Связь объектов" и с помощью специальных атрибутов родительского объекта. Рассмотрены преимущества и практические примеры использования каждого из этих способов. Приведено математическое описание модели, а также физическая модель таблиц базы данных, предназначенных для хранения гетерогенных информационных ресурсов.

Ключевые слова: интеграция, гетерогенные информационные ресурсы, гибкое управление связями

Введение

С переходом российских вузов на Федеральные государственные образовательные стандарты 3-го поколения (далее ФГОС-3) перед каждым вузом встает задача разработки основных образовательных программ (ООП). В соответствии с требованиями нового стандарта все ООП вуза должны быть обеспечены учебно-методической документацией и содержательными материалами по всем учебным курсам, дисциплинам (модулям). Содержание каждой из таких учебных дисциплин (модулей) должно быть представлено в сети Интернет или локальной сети образовательного учреждения [1].

Все содержательные образовательные элементы дисциплины в электронной форме должны быть обеспечены информационными ресурсами (ИР). Унаследованное хранилище ИР в том или ином виде существует в любом вузе. Повторное использование его содержимого и интеграция его с образовательными элементами дисциплин в настоящее время является очень актуальной задачей для всех вузов. Именно для решения этой задачи и предназначена данная модель.

Для интеграции образовательных элементов в электронной среде данная модель использует модель на основе интегрированных баз данных [2],

в которой отсутствует глобальная схема, связывающая источники данных (в отличие, например, от модели на основе мультибаз данных [3]), каждый источник представляется в структурированном виде, а связи устанавливаются непосредственно между источниками. Такая схема поддерживает децентрализованное управление, предоставляя локальным базам больше возможности управления разделяемой информацией. Степень интеграции не обязана быть полной, как в случае с глобальной схемой, — она зависит от потребностей пользователей, и, соответственно, такая система может быть либо сильно связанной, либо слабо связанной [4].

Для повторного использования основных образовательных элементов и представления их в сети Интернет или локальной сети образовательного учреждения необходимо их все хранить в базе данных (БД). Применение существующих традиционных подходов к проектированию структуры БД [5], в которых каждая сущность предметной области (в нашем случае это образовательный элемент) для хранения своих экземпляров представлена отдельной таблицей в БД, когда большинство атрибутов этой сущности являются столбцами этой таблицы и для некоторых атрибутов в процессе нормализации таблиц выделяются индивидуальные таблицы, а связи между сущностями реализуются с помощью связей

между столбцами таблиц (с использованием внешних ключей), нерационально для разработки структуры БД по хранению образовательных элементов вуза. Главной причиной является то, что вузу приходится управлять огромным числом образовательных элементов (учебными планами, дисциплинами, модулями, темами и т. д.), которые имеют достаточно сложные связи между собой, а число их атрибутов (названий, часов, семестров, форм контроля и т. д.), как и самих элементов, постоянно изменяется. Использование классической методики проектирования БД потребовало бы ее постоянной реорганизации каждый раз при добавлении новых образовательных элементов и их атрибутов и со временем привело бы к неограниченному разрастанию числа таблиц и их столбцов в БД.

В связи с этим возникает задача по проектированию структуры БД, которая наиболее рационально подходила бы для хранения образовательных элементов вуза, их атрибутов и связей между ними, обеспечивала бы не только принципиальное единообразие хранения всех данных, но и гибкость в управлении системой ограничений, накладываемых на значения атрибутов и связей между образовательными элементами. Представим математическое описание модели интеграции и хранения образовательных элементов.

Математическое описание модели интеграции и хранения образовательных элементов

Каждый образовательный элемент модели должен быть отдельным уникальным объектом в БД для того, чтобы затем можно было хранить его атрибуты и устанавливать связи данного объекта с другими объектами. Обозначим множество объектов

$$O = \{o_1, o_2, \dots, o_{n_o}\},$$

где n_o — общее число объектов в модели.

Примерами объектов в БД могут быть конкретные ООП вуза, их конкретные учебные планы, конкретные дисциплины, конкретные модули, конкретные темы, конкретные ИР и т. д. Сходные объекты модели можно объединить в соответствующие типы объектов. Обозначим множество типов объектов переменной

$$T = \{t_1, t_2, \dots, t_{n_t}\},$$

где n_t — общее число типов объектов в модели.

Каждый объект модели может относиться только к одному типу объектов (например, объект, который является дисциплиной, должен относиться только к типу объектов "Дисциплины"). Но каждый тип объектов может иметь несколько объектов (в вузе существует множество конкретных ООП, конкретных учебных планов, конкретных дисциплин, кон-

кретных модулей, конкретных тем и т. д.). Обозначим множество объектов i -го типа как ot_i :

$$ot_i = \{o_1, o_2, \dots, o_{n_{ot_i}}\} \& \sum_{i=1}^{n_t} n_{ot_i} = n_o,$$

где n_{ot_i} — общее число объектов типа ot_i .

Для того чтобы различать объекты одного типа между собой, необходимо хранить характеристики данных объектов, которые в БД могут быть представлены в виде атрибутов с их определенными значениями. Обозначим множество атрибутов объектов как A , тогда i -й атрибут будет обозначаться как a_i :

$$A = \{a_1, a_2, \dots, a_{n_a}\},$$

где n_a — общее число атрибутов объектов в модели.

Примерами атрибутов объектов могут быть название (например, названия дисциплин, модулей, тем), URL (например, на информационный ресурс в файловом хранилище вуза или на внешний источник в сети Интернет), семестр (например, дисциплины в рамках учебного плана) и т. д. Некоторые атрибуты объектов могут иметь только строго определенные значения из множества допустимых значений (домена). Например, семестр дисциплины в рамках учебного плана по ООП бакалавриата может быть любым целым числом из отрезка [1; 8]. Обозначим множество допустимых значений атрибута a_j как $D(a_j)$, тогда j -е допустимое значение атрибута a_j будет обозначаться как $d(a_{ij})$:

$$D(a_j) = \{d(a_{i1}), d(a_{i2}), \dots, d(a_{in_{D(a_j)}})\},$$

где $n_{D(a_j)}$ — общее число допустимых значений атрибута a_j .

Одним из источников ограничений на допустимые значения атрибутов объектов (например, на названия дисциплин) являются требования ФГОС-3, которые также определяют и некоторые типы объектов (учебные планы, программы практик и итоговой аттестации, дисциплин и т. д.).

Значение атрибута a_j для объекта o_i может быть записано как

$$o_i(a_j) = v_{ij},$$

где $i \in [1; n_o], j \in [1; n_a]$.

Некоторые объекты модели могут иметь несколько значений для одного атрибута. k -е значение атрибута a_j для объекта o_i может быть записано в виде

$$(o_i(a_j))_k = v_{ijk},$$

где $i \in [1; n_o], j \in [1; n_a]$ и $k \in \mathbb{N}$.

Примером атрибута с множественным значением является студенческая группа для учебного плана, так как по одному учебному плану в институте могут

учиться несколько групп студентов, и число таких групп может быть различно для каждого учебного плана.

Все атрибуты объекта можно также разделить на следующие рода в зависимости от типа данных их значения:

- целочисленные атрибуты; обозначим множество целочисленных атрибутов объектов как IA :

$$IA = \{a_1, a_2, \dots, a_{n_{ia}}\},$$

где n_{ia} — общее число целочисленных атрибутов объектов. Целочисленные атрибуты используются для хранения такой информации, как семестр и зачетные единицы трудоемкости (ЗЕТ) дисциплины в рамках учебного плана, года начала обучения студентов по данному учебному плану и т. д.;

- строковые атрибуты; обозначим множество строковых атрибутов объектов как SA :

$$SA = \{a_1, a_2, \dots, a_{n_{sa}}\},$$

где n_{sa} — общее число строковых атрибутов объектов. Строковые атрибуты используются для хранения названий ООП, учебных планов, дисциплин, модулей, тем и т. д.;

- атрибуты, содержащие дату и время; обозначим множество атрибутов объектов, содержащие дату и время как DA :

$$DA = \{a_1, a_2, \dots, a_{n_{da}}\},$$

где n_{da} — общее число атрибутов объектов, содержащих дату и время. Атрибуты, содержащие дату и время, могут использоваться, например, для хранения даты и времени создания учебных планов, дисциплин, модулей и т. д.;

- вещественные атрибуты; обозначим множество вещественных атрибутов объектов как RA :

$$RA = \{a_1, a_2, \dots, a_{n_{ra}}\},$$

где n_{ra} — общее число вещественных атрибутов объектов. Вещественные атрибуты используются для хранения процента обеспеченности дисциплин, модулей, тем минимальным набором ИР, необходимым в соответствии со структурой модели содержательных образовательных элементов дисциплины. Также вещественные атрибуты используются для хранения баллов студентов.

Один и тот же атрибут объекта может иметь только один род (например, семестр дисциплины может быть только целого типа, а ее название — только строкового). Каждый род атрибутов можно классифицировать на отдельные виды по следующим признакам:

- по существованию для объекта:
 - обязательные — это те атрибуты, которые должны иметь хотя бы одно значение для объектов некоторого типа объектов:

a_i , где $i \in [1; n_a]$ — обязательный атрибут для объектов типа t_j , где $j \in [1; n_d]$, если $\forall k \in [1; n_{ot_j}] \exists v_{ki} = o_k(a_i) \in D(a_i)$;

- абсолютно обязательные — это те атрибуты, которые должны иметь хотя бы одно значение для объектов всех типов объектов, т. е. для всех объектов:

a_i , где $i \in [1; n_a]$ — абсолютно обязательный атрибут для объектов, если $\forall k \in [1; n_o] \exists v_{ki} = o_k(a_i) \in D(a_i)$;

- необязательные — это те атрибуты, которые должны отсутствовать хотя бы у одного типа объектов:

a_i , где $i \in [1; n_a]$ — необязательный атрибут для объектов типа t_j , где $j \in [1; n_d]$, если $\exists k_1 \in [1; n_{ot_j}] \nexists v_{k_1 i} = o_{k_1}(a_i) \& \exists k_2 \in [1; n_{ot_m}]$,

$m \in [1; n_t] \& m \neq j: \exists v_{k_2 i} = o_{k_2}(a_i) \in D(a_i)$;

- по определенности значения:
 - с произвольным (неопределенным) значением — это те атрибуты, домен которых для каждого объекта может состоять из любых, заранее неопределенных по значению, но определенных по типу данных, элементов, и мощность (размерность) такого домена не ограничена:

a_i , где $i \in [1; n_a]$ — атрибут с произвольным значением для объектов, если $\forall k \in [1; n_o] v_{ki}$ заранее неизвестно $\& |D(a_i)| > N$, где $N \rightarrow \infty$;

- с определенным значением (или со значением из множества допустимых значений) — это те атрибуты, домен которых является конечным множеством со строго определенными заранее элементами:

a_i , где $i \in [1; n_a]$ — атрибут с определенным значением для объектов, если $\forall k \in [1; n_o] v_{ki} \in D(a_i) = \{d_{1i}, d_{2i}, \dots, d_{Ni}\}$, $N \in \mathbb{N}$;

- по единственности значения:
 - с единственным значением — это те атрибуты, которые должны иметь не более одного значения для объекта:

a_i , где $i \in [1; n_a]$ — атрибут с единственным значением для объектов, если $\forall k \in [1; n_o] \exists! v_{ki} \in D(a_i)$;

- с множественным значением — это те атрибуты, которые могут иметь более двух различных значений для объекта:

a_i , где $i \in [1; n_a]$ — атрибут с множественным значением для объектов, если $\forall k \in [1; n_o] \exists v_{kij_1} \in D(a_i) \& \exists v_{kij_2} \in D(a_i) \& \dots \exists v_{kij_N} \in D(a_i): v_{kij_1} \neq v_{kij_2} \neq \dots \neq v_{kij_N} \& N \in \mathbb{N}$;

- по характеру создания:
 - пользовательские — это те атрибуты, значения которых задаются пользователями;
 - системные — это те атрибуты, значения которых задаются автоматически.

Соответствие типов объектов и их атрибутов

	Атрибут 1	Атрибут 2	...	Атрибут M
Тип объекта 1	1	0	0	1
Тип объекта 2	0	1	0	1
...
Тип объекта N	0	1	0	0

Один и тот же атрибут может иметь несколько видов, и наборы видов атрибутов могут различаться в зависимости от объекта, к которому относится данный атрибут. Абсолютно обязательными для всех объектов являются дата создания объекта и информация о том, кем он был создан. Также для большинства объектов обязательным атрибутом является название. Остальные объекты в зависимости от типа объектов, к которому они принадлежат, имеют различные атрибуты. Связи типов объектов с их атрибутами могут быть представлены в виде бинарной матрицы размера $N \times M$ (табл. 1), строки которой будут соответствовать типам объектов, а столбцы — атрибутам объектов данного типа. Если на пересечении i -го типа объекта и j -го атрибута стоит цифра 1, то i -й тип объекта может иметь j -й атрибут.

Значения данной матрицы могут быть записаны в виде упорядоченных пар типов объектов и их атрибутов:

$$ToA = \{(t_i, a_j) | t_i \in T \& i \in [1; n_t], a_j \in A \& j \in [1; n_a]\}.$$

Таким образом, вышеизложенный подход позволяет обеспечить принципиальное единообразие хранения всех данных, а также за счет введения классификации атрибутов объектов по родам и видам обеспечить гибкость в управлении системой ограничений, накладываемых на значения атрибутов.

Среди всех типов объектов в модели стоит выделить тип объекта "Связь объектов". С помощью него имеется возможность указать отношения между объектами. Чаще всего в данной модели используется отношение иерархии объектов. В этом случае один из этих объектов является главным и носит название "Родительский объект", а другой объект является подчиненным и носит название "Дочерний объект". Для сохранения отношения иерархии данных объектов создается объект типа "Связь объектов" с двумя атрибутами, которые являются целочисленными (род атрибута), абсолютно обязательными (вид атрибута), с определенным единственным значением (виды атрибута) и системными (вид атрибута). Первый атрибут этого объекта хранит значение идентификатора (ID) родительского объекта (далее "ID родительского объекта"), второй — идентификатора дочернего объекта (далее "ID дочернего объекта"). Для установления отношения подчинения между одним родительским объектом и несколькими дочерними объектами необходимо создать столько же объектов типа "Связь объектов", сколько и дочерних объектов. Если при этом необходимо указать порядок

(приоритет) иерархии каждого дочернего объекта по отношению к родительскому, то для объекта типа связь необходимо указать атрибут "Сортировка дочернего элемента в рамках родительского". Объект типа "Связь объектов" можно использовать, например, для связи дисциплины в качестве родительского объекта и модулей в качестве дочерних объектов (рис. 1). Для каждой такой связи необходимо указать порядок подчинения, так как в учебном процессе важна последовательность изучения модулей друг за другом.

У объекта "Связь объектов", как у обычного объекта, могут быть другие атрибуты. Например, если он связывает объект типа "Учебный план" (родительский объект) и объект типа "Дисциплина" (дочерний объект), то в качестве его атрибутов могут быть семестр этой дисциплины в учебном плане, ее ЗЕТ, форма контроля и т. д. (рис. 2).

Отношения иерархии типов объектов друг с другом могут быть представлены в виде бинарной

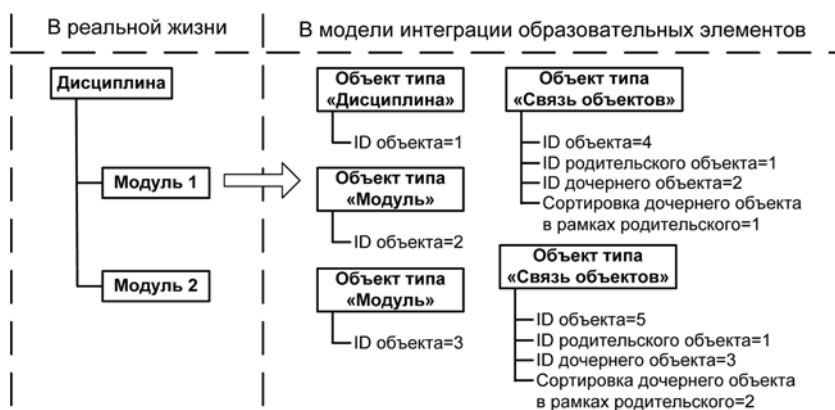


Рис. 1. Отношение между объектами в реальной жизни и в модели интеграции образовательных элементов



Рис. 2. Пример атрибутов объекта типа "Связь объектов"

Таблица 2

Отношения иерархии между типами объектов

	Тип объекта 1	Тип объекта 2	...	Тип объекта N
Тип объекта 1	0	1	0	0
Тип объекта 2	0	0	0	0
...
Тип объекта N	1	0	0	0

матрицы смежности размера $N \times N$ (табл. 2), строки и столбцы которой будут соответствовать типам объектов. Если на пересечении i -го типа объекта и j -го типа объекта ($i \neq j$) стоит цифра 1, то i -й тип объекта является родительским, а j -й — дочерним типом объектов.

Значения данной матрицы могут быть записаны в виде упорядоченных пар типов объектов:

$$ToTo = \{(t_i, t_j) | t_i, t_j \in T \ \& \ i, j \in [1; n_i]\}.$$

Таким образом, использование дополнительного объекта типа "Связь объектов" обеспечивает гибкость в управлении системой ограничений, накладываемых на связи между образовательными элементами — объектами. В этом случае при установлении связей между объектами имеется возможность указания приоритета родства дочерних объектов по отношению к родительским, а также любых других атрибутов связей объектов друг с другом.

В случае отсутствия необходимости в управлении сложными связями и ограничениями в предлагаемой модели связь типов объектов может осуществляться по классической схеме с помощью специальных атрибутов родительского объекта, в которых будут храниться ID дочерних объектов. Данные атрибуты должны быть целочисленными, системными и с определенным множественным значением. Достоинством данного способа связей объектов друг с другом является экономия места за счет того, что нет необходимости дополнительно создавать объекты типа "Связь объектов" и указывать для них атрибуты "ID родительского объекта" и "ID дочернего объекта".

Физическая модель таблиц БД, предназначенных для хранения гетерогенных информационных ресурсов

Спроецируем математическое описание модели интеграции образовательных элементов на физическую (дatalogическую) модель таблиц в БД. Анализ существующих моделей таблиц БД показал, что для нашего случая хорошо подходит модель хранения данных EAV (Entity-Attribute-Value) [6]. Модель EAV представляет собой некую универсальную структуру описания данных, которая в самом общем смысле подразумевает хранение всех данных в виде так называемых таблиц-справочников: справочника сущностей, справочника атрибутов и спра-

вочника значений [7]. В данной модели для хранения различных типов данных (строковых, числовых и т. д.) выделяются отдельные таблицы-справочники значений, в простейшем случае используется одна таблица с применением наиболее общего формата данных. Справочник значений помимо хранения всех данных обеспечивает взаимосвязь с остальными таблицами-справочниками, поскольку содержит поля метаданных для ссылки на справочники сущностей и атрибутов, при этом каждый из атрибутов определенной сущности отображается на одну запись в справочнике значений. Таким образом, весь массив хранимых данных, образованный совокупностью значений таблиц-справочников, организуется фактически в один линейный список, что обеспечивает принципиальное единообразие хранения всех данных, а следовательно, и существенную гибкость (настраиваемость) системы, разрабатываемой на основе данной модели.

Вся информация в БД, имеющей структуру модели EAV, хранится в пяти таблицах (рис. 3).

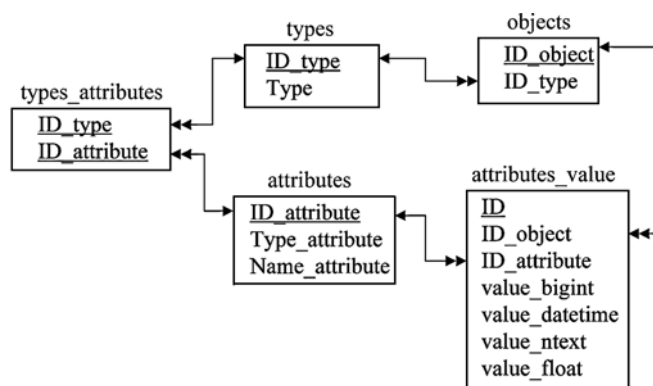


Рис. 3. Схема таблицы БД для хранения модели интеграции образовательных элементов

Таблица *types* содержит все типы объектов (ООП, учебные планы, дисциплины, модули, темы и т. д.). Все объекты, а именно их уникальные идентификаторы, содержатся в таблице *objects*. Все атрибуты объектов описаны в таблице *attributes*. Связи типов объектов с атрибутами описаны в таблице *types_attributes*. Значения атрибутов объектов находятся в таблице *attributes_values*.

Заключение

Рассмотренная модель интеграции образовательных элементов может быть обобщена до модели интеграции любых гетерогенных объектов окружающего мира. Эта более общая модель будет позволять хранить любые атрибуты этих объектов, а также различные виды связей между ними. Для хранения информации о данных гетерогенных объектах также можно будет использовать представленную выше структуру БД, использующей EAV-модель хранения данных.

Список литературы

1. **Федеральный** государственный образовательный стандарт высшего образования (макет). URL: <http://cme.donstu.ru/index.php?action=pages&id=104>
2. **Elmagarmid A., Rusinkiewicz M., Sheth A.** Management of Heterogeneous and Autonomous Database Systems. San Francisco, California: Morgan Kaufmann Publishers, 1998.
3. **Litwin W.** From database to multidatabase systems: Why and how // British National Conference on Databases. Cambridge Press, 1988.

4. **Кашников А. В., Лядова Л. Н.** Интеграция гетерогенных источников данных на основе рекурсивной декомпозиции // Информационные технологии и знания. 2011. Т. 5, № 3.
5. **Кириллов В. В.** Основы проектирования реляционных баз данных. СПб.: ИТМО, 1997. URL: <http://khpi-ii.p.mpk.kharkiv.edu/library/dbms/kirl/>
6. URL: http://en.wikipedia.org/wiki/Entity-attribute-value_model
7. **Nadkarni P. M., Marengo L., Chen R., Skoufos E., Shepherd G., Miller P.** The EAV/CR Model of Data Representation // J. Am. Med. Inform. Assoc. 2000. URL: http://ycmi.med.yale.edu/nadkarni/eav_CR_contents.htm

I. G. Ignatova, Vice-rector for Education, **E. O. Bereza**, Head of Department, National Research University of Electronic Technology (MIET), e-mail: igi@mice.ru

The Model of Heterogeneous Information Resources Integration with Flexible Communication Management

The model of integration and storage of heterogeneous information resources as objects, their attributes and the relationships between them provides the fundamental uniformity of data storage as well as flexibility in the management of constraints on the attributes values of objects and relationships between them are considered. On the example of compliance with the requirements of federal state educational standard of third generation to ensure basic educational programs instructional documentation and informative materials demonstrates the use of this model. Classification of object attributes model for genera and species in the context of certain evidence are provided. Two ways flexible relationship management between heterogeneous information resources: using a special object of type "Communication objects" and using the special attributes of the parent objects are described. Advantages and practical examples of the use of each of these methods are considered. The mathematical description of the model, physical model of the database tables for storage of heterogeneous information resources are provided.

Keywords: integration, heterogeneous information resources, flexible communication management

References

1. **Federalnyiy** gosudarstvennyiy obrazovatelnyiy standart vyishego obrazovaniya (maket). URL: <http://cme.donstu.ru/index.php?action=pages&id=104>.
2. **Elmagarmid A., Rusinkiewicz M., Sheth A.** Management of Heterogeneous and Autonomous Database Systems. San Francisco, California: Morgan Kaufmann Publishers, 1998.
3. **Litwin W.** From database to multidatabase systems: Why and how // British National Conference on Databases. Cambridge Press, 1988.

4. **Kashnikov A. V., Lyadova L. N.** Integratsiya geterogennykh istochnikov dannykh na osnove rekursivnoy dekompozitsii. *Informatsionnye tekhnologii i znaniya*. 2011. V. 5, N 3.
5. **Kirilov V. V.** Osnovyi proektirovaniya relyatsionnykh baz dannykh, SPb.: ITMO, 1997. URL: <http://khpi-ii.p.mpk.kharkiv.edu/library/dbms/kirl/>
6. URL: http://en.wikipedia.org/wiki/Entity-attribute-value_model
7. **Nadkarni P. M., Marengo L., Chen R., Skoufos E., Shepherd G., Miller P.** The EAV/CR Model of Data Representation. *J. Am. Med. Inform. Assoc.* 2000. URL: http://ycmi.med.yale.edu/nadkarni/eav_CR_contents.htm

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В БИМЕДИЦИНСКИХ СИСТЕМАХ INFORMATION TECHNOLOGIES IN BIOMEDICAL SYSTEMS

УДК 004.932.2

Д. М. Михайлов, канд. техн. наук, доц., e-mail: dmmikhajlov@mephi.ru,

А. В. Стариковский, ассистент, e-mail: avstarikovskij@mephi.ru,

В. Н. Конев, инженер, e-mail: konev.vn@gmail.com,

А. С. Смирнов, инженер, e-mail: smirnov_as@bk.ru,

А. Ю. Боручинкин, инженер, e-mail: boruchinkin28@ya.ru,

С. М. Харьков, инженер, e-mail: strategys@yandex.ru,

Национальный исследовательский ядерный университет "МИФИ", Москва

Программное обеспечение для автоматизированного анализа изображений желудочно-кишечного тракта*

Данная статья посвящена разработке программного обеспечения, позволяющего в автоматизированном режиме обрабатывать изображения желудочно-кишечного тракта пациента, полученные беспроводной эндоскопической капсулой. Приводится описание архитектуры программного обеспечения для анализа изображений пищеварительного тракта, входящего в состав капсульного эндоскопического комплекса "Ландыш", а также результаты его тестирования.

Ключевые слова: программное обеспечение автоматизированного рабочего места, распознавание патологий желудочно-кишечного тракта, обработка изображений, красный канал, классификация изображений, абстрактный класс, обработка видеоотчета, эндоскопическая капсула

Введение

Беспроводная капсульная эндоскопия — это современное направление обследования пищеварительного тракта пациента, позволяющее с помощью миниатюрной проглатываемой капсулы получать изображения желудочно-кишечного тракта (ЖКТ) пациента. Данные изображения далее передаются на считывающее устройство, затем на персональный компьютер врача, на котором с помощью специального программного обеспечения проводится автоматическая обработка изображений. Таким образом, программное обеспечение играет особую роль в капсульных эндоскопических комплексах, позволяя в значительной степени убыстрить процесс постановки диагноза [1—3].

Программное обеспечение автоматизированного рабочего места (ПО АРМ) врача предназначено для автоматизированного анализа изображений ЖКТ, полученных эндоскопической капсулой, и после-

дующего формирования проектов медицинских заключений по результатам обследования пациента.

Вопросу автоматизированного анализа изображений ЖКТ уделяется большое внимание в различных научных трудах [4—7], так как программное обеспечение, распознающее патологии пищеварительного тракта, способно значительно сократить время, затрачиваемое гастроэнтерологом на анализ потока изображений, полученных с беспроводной капсулы [8]. Существующие решения в данной области имеют свои плюсы и минусы, поэтому было решено разработать собственное ПО, отвечающее конкретным требованиям разрабатываемого эндоскопического комплекса и лишенное недостатков известных аналогов.

Основные характеристики ПО АРМ

Предлагаемое ПО АРМ врача обеспечивает:

- систематизированное хранение и обработку данных о пациентах;
- автоматизированный анализ полученных данных;
- просмотр и обработку результатов анализа;
- предварительное формирование в автоматизированном режиме заключения по результатам обследования.

* Публикация подготовлена по результатам работ, выполненных НИЯУ МИФИ и ОАО РТИ в рамках договора № 02.G25.31.0018 от 12 февраля 2013 года между Открытым акционерным обществом "Радиотехнический институт имени академика А. Л. Минца" и Министерством образования и науки Российской Федерации.

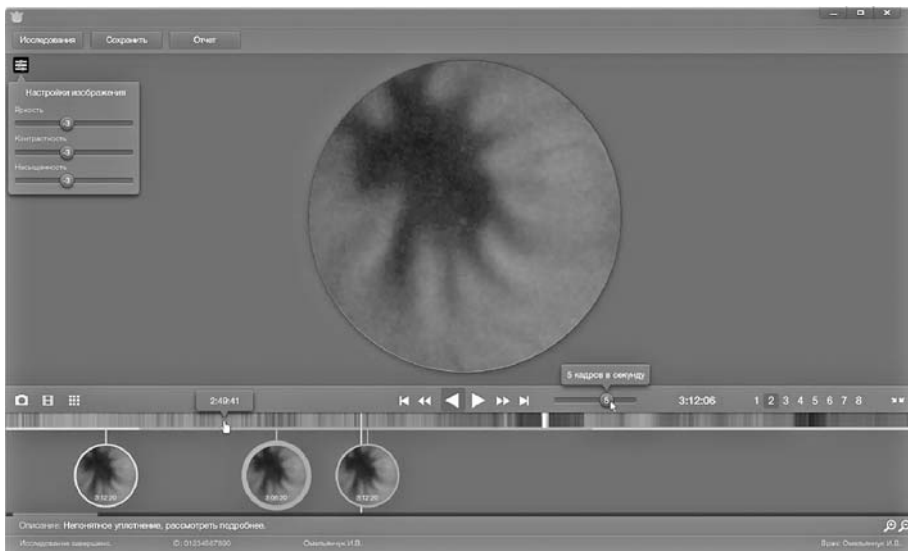


Рис. 1. Видеоотчет по итогам исследования



Рис. 2. Вывод изображения на экран

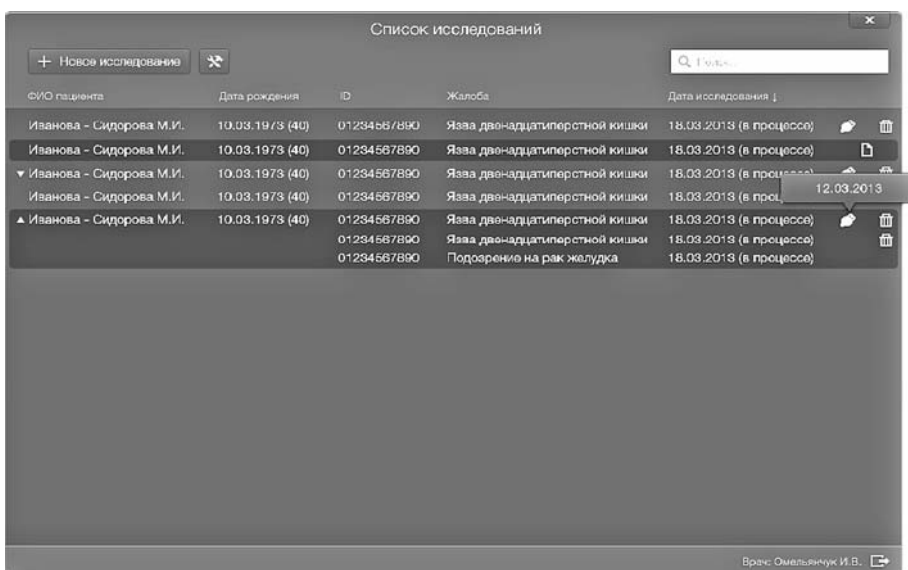


Рис. 3. Интерфейс ПО АРМ врача

ПО АРМ выявляет кадры с аномальными изображениями — распознанной патологией и подозрениями на патологию на основе их анализа в соответствии с созданным полным атласом патологий всего ЖКТ, определяет проблемные точки в организме, такие как опухоли, язвы, кровотечения и полипы. Далее ПО АРМ врача составляет видеоотчет с наиболее важными с точки зрения постановки диагноза кадрами (рис. 1), что значительно сокращает временные затраты и упрощает работу врача по формированию заключения о состоянии пациента.

Стоит отметить, что ПО АРМ врача, в отличие от мировых аналогов [8, 9], разрабатывается с функциями автоматизированного обнаружения аномалий цвета, формы и текстуры [10], редактирования получаемого изображения (маркировка, изменение размеров, масштабирование, улучшение изображения), что повышает эффективность работы специалиста в ходе анализа материалов исследования. Также ПО АРМ врача позволяет выводить на экран до 15 снимков ЖКТ пациента (рис. 2) и выполнять любой тип записи отчета.

На рис. 3. изображен интерфейс ПО АРМ врача, демонстрирующий список пациентов, проходящих исследование.

ПО АРМ врача имеет возможность обработки данных, полученных от других капсульных эндоскопических комплексов за счет поддержки работы со стандартным форматом изображений JPEG2000.

Максимальное время, требуемое для проведения первичной обработки видеоинформации одного исследования и автоматизированного формирования предварительного отчета для врача, — около 10 мин. Среднее время обработки одного исследования врачом с возможностью формирования соответствующего отчета не превышает 5 мин [11].

ПО АРМ врача функционирует под управлением операционной системы Microsoft Windows версий Vista, XP, 7, 8 и обеспечено защитой от несанкционированного копирования и эксплуатации посредством привязки каждой копии к регистрационному ключу [11].

Разработка архитектуры ПО АРМ

Рассмотрим процесс разработки архитектуры программного обеспечения обработки изображений ЖКТ. Программное обеспечение для беспроводной эндоскопической капсулы должно выполнять три последовательных действия:

- считывать изображения желудочно-кишечного тракта из абстрактного хранилища (база данных, видео, файловая система);
- вычислять формальные признаки, указывающие на возможное наличие патологий;
- подавать результат на вход выбранному алгоритму машинного обучения.

Самостоятельно реализовывать алгоритм машинного обучения нет необходимости, потому что для этой задачи существуют готовые решения [12].

В основе ПО лежат три класса: Bank, Feature и Output.

Bank — абстрактный класс-итератор, объявляет два метода, свойственные итераторам: `HasNextItem` и `GetNextItem`. Для различных источников эти два метода могут выполнять разные функции, например, если источником является видеопоток, то `HasNextItem` будет возвращать `true`, если в потоке еще есть кадры, в противном случае — `false`, и тогда `GetNextItem` будет возвращать очередной кадр. Объектом, который возвращает `GetNextItem`, является объект класса `BankItem`, имеющий два поля: изображение и описание изображения. Описание изображения — это таблица со строковыми ключами и их значениями.

Feature — базовый абстрактный класс, определяющий только один метод — `GetValue`, который принимает изображение и вычисляет по нему набор формальных признаков. Общим для большинства признаков является то, что они имеют большое число конфигурационных параметров, например, для признака "гистограмма канала" это цветовая модель, название канала, число ячеек гистограммы. Для однородной работы с конфигурационными параметрами существует класс-таблица `FeatureConfiguration` со строковыми ключами и числовыми значениями.

Output также является базовым абстрактным классом, который накапливает вычисленные признаки для всех изображений и преобразует результат в соответствии с форматом, который требует вы-

бранный алгоритм машинного обучения. За два описанных шага отвечают методы `Append` и `Format`.

Для того чтобы выбрать алгоритм машинного обучения (наивный байесовский классификатор, линейная регрессия, логистическая регрессия или машина опорных векторов), нужно определить показатели, по которым эти алгоритмы можно сравнивать.

Обратимся к математическому аппарату.

Дано: вход X —выход Y .

Наблюдаем последовательность реализаций (пар): $(x_1, y_1), (x_2, y_2)$ и т. д.

(X_n, Y_n) — обучающая выборка. Зависимость $Y = a(X)$ такова, что предсказанные значения $Y^* = a(X)$ как можно более точно аппроксимируют фактическое выходное значение Y , соответствующее входу X [13].

На математическом языке задачу классификации изображений на "аномальные" и "здоровые" можно описать следующим образом: существует множество X из N объектов и необходимо найти функцию $a(x_i) = y_i$, где $x_i \in X, y_i \in Y = \{0, 1\}^N$, где 0 означает, что x_i не стоит рассматривать в первую очередь, 1 — что стоит [13].

Таким образом, была собрана обучающая выборка X , в которой для каждого $x_i \in X$ известны значения $y_{correct_i}$.

X разделена на X_{test} и $X_{control}$, такие что $X_{test} \cup X_{control} = X$ и $X_{test} \cap X_{control} = \emptyset, |X_{test}| = N_{test}, |X_{control}| = N_{control}$.

Затем алгоритм на базе функции a обучался на X_{test} и запускался на $X_{control}$, таким образом, получалось два множества: $y_{control} = a(X_{control})$ и $y_{correct}$. Сравнивая их, можно строить оценки качества алгоритма:

- Classification accuracy (CA) — доля верно классифицированных изображений из обрабатываемого массива изображений:

$$CA = \frac{\sum [y_{control_i} = y_{correct_i}]}{N} \quad (1)$$

- Average Probability (AP) — если функция a дает не бинарный ответ, а оценивает вероятность, то можно посчитать среднее значение по тем вероятностям, которые присваивались правильно классифицированным изображениям. Этот показатель характеризует устойчивость a :

$$AP = \frac{\sum P(y_{correct_i})}{N} \quad (2)$$

- True Positives (TP) — число правильно классифицированных подозрительных изображений:

$$TP = \sum [y_{control_i} = 1][y_{correct_i} = 1] \quad (3)$$

- True Negative (*TN*) — число правильно классифицированных нормальных изображений:

$$TN = \sum [y_{control_i} = 0][y_{correct_i} = 0]. \quad (4)$$

- False Negative (*FN*) — число подозрительных изображений, которые были классифицированы, как нормальные. Это число должно быть как можно меньше, потому что это изображения, на которых потенциально есть признаки заболевания, но которые не увидит врач-эндоскопист:

$$FN = \sum [y_{control_i} = 0][y_{correct_i} = 1]. \quad (5)$$

- False Positive (*FP*) — число нормальных изображений, которые были классифицированы как подозрительные. Нужно стараться сократить это число, чтобы врачу не приходилось просматривать подготовленную выборку подозрительных изображений:

$$FP = \sum [y_{control_i} = 1][y_{correct_i} = 0]. \quad (6)$$

- *Recall* — доля подозрительных изображений, показанных врачу, по отношению ко всем подозрительным изображениям:

$$Recall = \frac{TP}{TP + FN}. \quad (7)$$

- *Precision* — доля подозрительных изображений, показанных врачу по отношению ко всем показанным изображениям:

$$Precision = \frac{TP}{TP + FP}. \quad (8)$$

- *F-measure* — среднегармоническое между *Precision* и *Recall*:

$$F = \frac{2PrecisionRecall}{Precision + Recall}. \quad (9)$$

Результаты проведенной оценки приведены в таблице.

Сравнение алгоритмов машинного обучения

Параметр	Наивный байесовский классификатор	Линейная регрессия	Логистическая регрессия	Машина опорных векторов
<i>CA</i>	0,929	0,748	0,789	0,791
<i>AP</i>	0,912	0,606	0,680	0,791
<i>TP</i>	46	49	48	49
<i>FP</i>	1	17	13	14
<i>FN</i>	4	1	2	1
<i>TN</i>	20	4	8	7
<i>Recall</i>	0,920	0,980	0,960	0,980
<i>Precision</i>	0,979	0,742	0,787	0,778
<i>F2</i>	0,948	0,845	0,865	0,867

Из таблицы видно, что самые лучшие показатели дает наивный байесовский классификатор. Однако такой результат связан с набором признаков, на котором тестировались классификаторы, и поэтому был выбран следующий по величине показателя *F2* алгоритм — машина опорных векторов.

В процессе выбора реализации алгоритма машинного обучения были рассмотрены три реализации машины опорных векторов: Orange, SVM-Light и LibSVM.

Orange. Библиотека Orange, кроме реализаций многих алгоритмов машинного обучения, включает в себя полноценный инструмент для анализа данных. Библиотека имеет интерфейс для языка Python, поэтому плохо встраивается в разрабатываемое программное обеспечение (на языке C++). Библиотека Orange позволяет воспользоваться сильными сторонами языка Python и очень просто вычислить основные показатели качества работы сразу нескольких алгоритмов машинного обучения. Показатели для таблицы были рассчитаны с помощью библиотеки Orange. Код для расчета этих показателей приведен ниже.

Код для генерации оптимизированной гистограммы красного канала:

```
29data = orange.ExampleTable('data.txt')

learners = [
    orange.BayesLearner(name = 'bayes'),
    orange.LinearLearner(name = 'linear'),
    orange.LogRegLearner(name = 'logistic'),
    orange.SVMLearner(name = 'svm'),
]

result = orngTest.crossValidation(learners, data)
CAs = orngStat.CA(result)
APs = orngStat.AP(result)
CMs = orngStat.confusionMatrices(result)

print learners[index].name,\
      CAs[index], APs[index],\
      CMs[index].TP, CMs[index].FP,\
      CMs[index].FN, CMs[index].TN,\
      orngStat.sens(CMs[index]),\
      orngStat.PPV(CMs[index]),\
      orngStat.F1(CMs[index]),\
```

Нужно отметить, что решение получается очень компактным.

SVMLight. Реализация SVMLight написана на языке C и включает в себя множество оптимизаций для ускорения вычислений, поэтому эта реализация хорошо подходит для работы с большими обучающими выборками. Утилита SVMLight предоставляет удобный консольный интерфейс, однако не очень хорошо подходит для встраивания в сторонний код.

LibSVM. LibSVM представляет собой набор библиотек и нескольких базовых утилит, использующих эти библиотеки. LibSVM, так же как SVMLight, написан на языке С и обладает хорошей производительностью. В то же время LibSVM значительно проще встроить в код, разрабатываемый на С или С++, чем SVMLight. Вместе с исходным кодом LibSVM распространяется утилита для оптимизации параметров модели и утилита для кроссвалидации.

Таким образом, в качестве реализации машины опорных векторов был выбран LibSVM.

Архитектура программного обеспечения, описанная выше, хорошо ложится на проблемную область. На рис. 4 показан абстрактный класс Bank и его наследники.

Наследник DirectoriesBank реализует логику простейшей иерархической базы данных. Предположим, что имеется файловая система, дерево которой представлено ниже:

```

healthy
  healthy01.jpg
  healthy02.jpg
bleeding
  bleeding01.png
  
```

Последовательно вызывая методы HasNextItem и GetNextItem класса DirectoriesBank, можно получить вывод:

```

32healthy01.jpg healthy
healthy02.jpg healthy
bleeding01.png bleeding
  
```

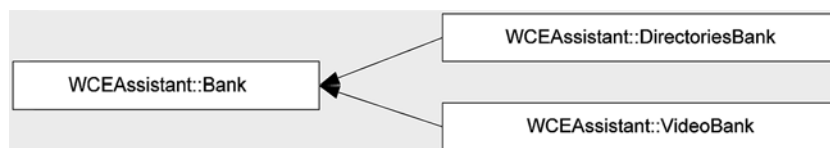


Рис. 4. Диаграмма наследования для абстрактного класса Bank

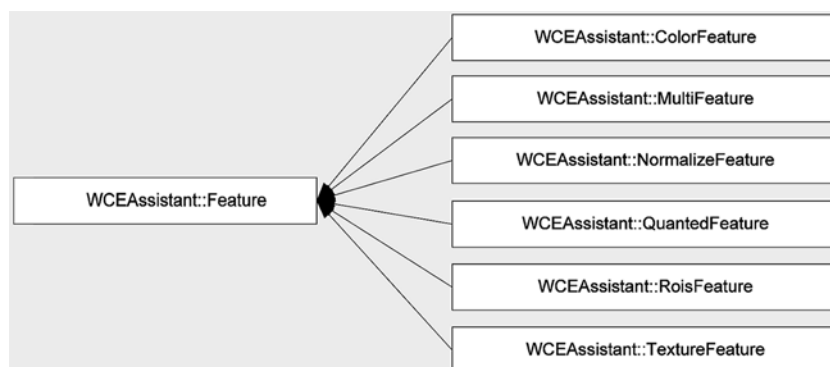


Рис. 5. Диаграмма наследования для абстрактного класса Feature

Такой подход очень удобен для хранения обучающей выборки. Файловая система может иметь вложенную структуру, а изображения могут представлять собой символичные ссылки друг на друга.

Наследник VideoBank предназначен для работы с видеопотоком. Методы HasNextItem и GetNextItem реализованы таким образом, что HasNextItem возвращает true до тех пор, пока не достигнут последний кадр видео; GetNextItem при каждом вызове возвращает новый кадр, а если видеопоток исчерпан, возвращается последний кадр.

На рис. 5 показан абстрактный класс Feature и его наследники.

Наследник ColorFeature реализует логику построения гистограммы канала изображения. Нужно заметить, что результатом работы является ненормализованная гистограмма.

Для нормализации используется класс NormalizeFeature.

Наследник TextureFeature служит для построения текстурной гистограммы. Логика работы этого класса такова, что для каждого изображения, поданного ему на обработку, класс считает около 30 сверток, что неизбежно сказывается отрицательно на производительности. Поэтому от использования этого модуля при обработке видео было решено отказаться.

Классы NormalizedFeature и QuantedFeature реализуют похожую функциональность в том смысле, что они преобразуют уже посчитанную гистограмму некоторой величины.

Классы RoisFeature и MultiFeature также похожи в том смысле, что результатом их работы является объединение значений признаков, посчитанных другими модулями.

LibSVMOutput — это единственный наследник, отвечающий за вывод признаков в формате, с которым работает библиотека LibSVM. Стандартная версия LibSVM работает с разреженными матрицами признаков, т. е. при описании обучающей выборки для некоторых объектов могут быть не указаны некоторые признаки.

Тестирование ПО АРМ. Для тестирования программного обеспечения была взята обучающая выборка из 110 изображений, полученных с эндоскопической капсулы, из которых 50 содержали здоровые участки желудочно-кишечного тракта и 60 — с признаками патологии. На этой выборке была проведена процедура обучения. Процесс повторялся для разных параметров алгоритма, лучший результат выбирался с помощью кроссвалидации. Рассмотрим результаты тестирования на примере гистограммы красного канала.

Код для вычисления гистограммы красного канала приведен ниже:

```
FeatureConfiguration configuration;  
configuration["Bins"] = 10;  
configuration["Optimize histogram"] = 0;  
configuration["Channel"] = 0; // Red  
ColorFeature red(configuration);  
NormalizedFeature redNormalized(red);  
FeatureValue feature =  
redNormalized.GetValue(image);
```

Результат поиска оптимальных параметров показан на рис. 6, где линиями показана локализация предполагаемого кровотечения.

Проверка соответствия характеристик альфа-версии ПО АРМ врача эндоскопического комплекса "Ландыш" второго поколения техническим требованиям выполнялась методом натуральных испытаний возможности:

- хранения и обработки данных;
- анализа видеозображения с результатами специального исследования, в котором предусмотрены признаки заболеваний;
- просмотра результатов анализа — кадров с аномальным изображением;
- формирования в автоматизированном режиме заключения по результатам обследования.

Тестирование проводилось на персональном компьютере со следующими характеристиками:

- процессор архитектуры x86 или x86_64 с тактовой частотой не менее 1 ГГц;
- объем оперативной памяти не менее 1 Гбайт;
- технология — DDR или выше;
- видеокарта — дискретная или встроенная, с объемом видеопамати, дискретной или разделяемой в ОЗУ, не менее 64 Мбайт;
- операционная система — Windows XP Home/Professional SP3, Windows XP 64 Home/Professional SP3, Windows Vista, Windows 7 x32/x64 SP1 Business/Home/Ultimate, Windows 8.

В результате проверки соответствия функциональных характеристик альфа-версии ПО АРМ врача эндоскопического комплекса "Ландыш" второго поколения техническим требованиям были проверены следующие функции: систематизированное хранение и обработка данных о пациентах, автоматизированный анализ полученных материалов, просмотр и обработка результатов анализа, предварительное формирование в автоматизированном режиме заключения по результатам обследования. Было выявлено, что функциональные характеристики ПО соответствуют техническим требованиям.

Заключение

Таким образом, было реализовано программное обеспечение, позволяющее проводить автоматизированный анализ изображений ЖКТ пациента, входящее в состав капсульного эндоскопического комплекса "Ландыш". Распознавание патологических изображений является одной из наиболее важных задач обработки изображений, полученных с эндоскопической капсулы. Оптимальным решением данной задачи является программное обеспечение, работающее на основе гистограммного анализа. Этот метод позволяет успешно проводить анализ или повышает эффективность анализа ЖКТ на основе автоматизированной компьютерной диагностики, повышая диагностическую значимость эндоскопического обследования.

В дальнейшем планируется создание виртуальной модели ЖКТ, представляющей новый способ хранения и анализа медицинской информации, и создание постоянно обновляемой базы данных обработанных изображений. Такая база данных сможет заполняться из множества удаленных компьютеров через сеть Интернет. На основе базы данных будет разработана экспертная система диагностики ЖКТ по данным эндоскопического обследования, которая может функционировать в онлайн-режиме.

Список литературы

1. **Iddan G., Meron G., Glukhovsky A., Swain P.** Wireless capsule endoscopy // *Nature*, 2000. V. 405, N. 6785. P. 417.
2. **Galmiche J. P., Coron E., Sacher-Huvelin S.** Recent developments in capsule endoscopy // *Gut*. 2008. N. 57. P. 695—703.
3. **Munoz-Navas M.** Capsule endoscopy // *World J. Gastroenterol.* 2009. N. 15(13). P. 1584—1586.
4. **Lecheng Y., Yuen P. C., Jianhuang L.** Ulcer detection in wireless capsule endoscopy images // *21st International Conference on Pattern Recognition (ICPR)*. 2012.
5. **Jebarani W. S. L., Daisy V. J.** Assessment of Crohn's disease lesions in Wireless Capsule Endoscopy images using SVM based classification // *International Conference on Signal Processing Image Processing & Pattern Recognition (ICSIPR)*. 2013.
6. **Chen D., Wang H., Hu C., Liu Z.** A novel strategy to label abnormalities for Wireless Capsule Endoscopy frames sequence // *IEEE International Conference on Information and Automation*. 2011.

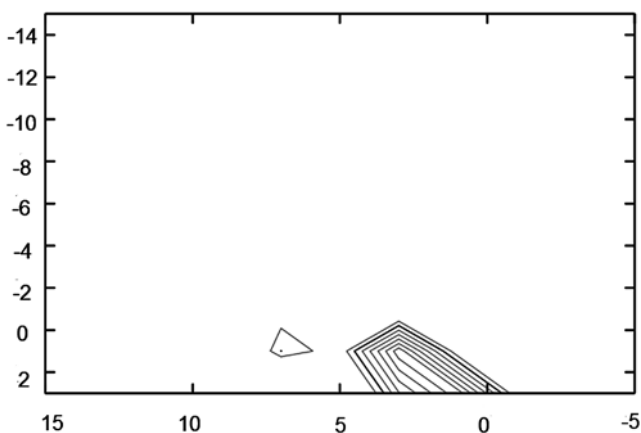


Рис. 6. Сетка поиска оптимальных параметров. Гистограмма красного канала

7. **Htwe T. M., Poh C. K., Li L., Liu J., Ong E. H., Yu Ho K.** Vision-based techniques for efficient Wireless Capsule Endoscopy examination // Defense Science Research Conference and Expo (DSR). 2011.

8. **Kukushkin A., Mikhaylov D., Ivanova E.** Recognition of Hemorrhage in the Images of Wireless Capsule Endoscopy // The 16th IEEE Mediterranean Electrotechnical Conference MELECON. 2012. P. 899–902.

9. **Михайлов Д. М., Хабибуллин Т. Р., Зуйков А. В., Толстая А. М.** Обзор сетевого функционала программного обеспечения автоматизированного рабочего места врача капсульных эндоскопических комплексов // Спецтехника и связь. 2013. № 4. С. 45–48.

10. **Mikhaylov D., Zhukov I., Konev V., Starikovskiy A., Khabibullin T., Tolstaya A., Kukushkin A.** Review of Features and Meta-

features Allowing Recognition of Abnormalities in the Images of GIT // 17th IEEE Mediterranean Electrotechnical Conference. Beirut, Lebanon. 2014.

11. **Клоков В. А., Жорин Ф. В., Андряков Д. А.** Автоматизированное рабочее место врача и разработка программного обеспечения для АРМ врача // Спецтехника и связь. 2013. № 4. С. 28–30.

12. **Ранетов А. М., Зуйков А. В., Кусакин И. И.** Обзор алгоритмов классификации изображений, полученных с помощью капсульного эндоскопа // Спецтехника и связь. 2013. № 4. С. 31–33.

13. **Айвазян С. А., Енюков И. С., Мешалкин Л. Д.** Основы моделирования и первичная обработка данных. М.: Финансы и статистика, 1983.

D. M. Mikhaylov, Associate Professor, e-mail: dmmikhajlov@mephi.ru,

A. V. Starikovskiy, Assistant, e-mail: avstarikovskij@mephi.ru,

V. N. Konev, Engineer, e-mail: konev.vn@gmail.com,

A. S. Smirnov, Engineer, e-mail: smirnov_as@bk.ru,

A. Yu. Boruchinkin, Engineer, e-mail: boruchinkin28@ya.ru,

S. M. Kharkov, Engineer, e-mail: strategys@yandex.ru,

National Research Nuclear University MEPHI

Software for the Automated Analysis of the Gastrointestinal Tract Images

The wireless capsule endoscopy is a new method of the gastrointestinal tract examination providing fast and comfortable medical testing of the entire digestive tract. Special software allows automated processing of data identifying images with pathologies or suspected abnormality. This article is devoted to the creation of the software allowing to automatically process the patient's gastrointestinal tract images which are received with the help of wireless endoscopic capsule. In this article the description of the software architecture for the gastrointestinal tract images processing which is a part of the "Landysh" capsular endoscopic complex is offered. The article also contains comparison of machine-learning algorithms as well as the information about the results of this software testing.

Keywords: software of the automated workplace, detection of digestive tract pathologies, image processing, red channel, classification of images, abstract class, processing video-report, endoscopic capsule

References

1. **Iddan G., Meron G., Glukhovskiy A., Swain P.** Wireless capsule endoscopy. *Nature*. 2000. V. 405, N. 6785. P. 417.

2. **Galmiche J. P., Coron E., Sacher-Huvelin S.** Recent developments in capsule endoscopy. *Gut*. 2008. N. 57. P. 695–703.

3. **Munoz-Navas M.** Capsule endoscopy. *World J. Gastroenterol*. 2009. N. 15(13). P. 1584–1586.

4. **Lecheng Y., Yuen P. C., Jianhuang L.** Ulcer detection in wireless capsule endoscopy images. *21st International Conference on Pattern Recognition (ICPR)*. 2012.

5. **Jebarani W. S. L., Daisy V. J.** Assessment of Crohn's disease lesions in Wireless Capsule Endoscopy images using SVM based classification. *International Conference on Signal Processing Image Processing & Pattern Recognition (ICSIPR)*. 2013.

6. **Chen D., Wang H., Hu C., Liu Z.** A novel strategy to label abnormalities for Wireless Capsule Endoscopy frames sequence. *IEEE International Conference on Information and Automation*. 2011.

7. **Htwe T. M., Poh C. K., Li L., Liu J., Ong E.H., Yu Ho K.** Vision-based techniques for efficient Wireless Capsule Endoscopy examination. *Defense Science Research Conference and Expo (DSR)*. 2011.

8. **Kukushkin A., Mikhaylov D., Ivanova E.** Recognition of Hemorrhage in the Images of Wireless Capsule Endoscopy. *The 16th IEEE Mediterranean Electrotechnical Conference MELECON*. 2012. P. 899–902.

9. **Mikhaylov D. M., Khabibullin T. R., Zuykov A. V., Tolstaya A. M.** Обзор сетевого функционала программного обеспечения автоматизированного рабочего места врача капсульных эндоскопических комплексов. *Спецтехника и связь*. 2013. N. 4. P.45–48.

10. **Mikhaylov D., Zhukov I., Konev V., Starikovskiy A., Khabibullin T., Tolstaya A., Kukushkin A.** Review of Features and Meta-features Allowing Recognition of Abnormalities in the Images of GIT. *17th IEEE Mediterranean Electrotechnical Conference*. Beirut, Lebanon. 2014.

11. **Klokov V. A., Zhorin F. V., Andryakov D. A.** Avtomatizirovannoe rabochee mesto vracha i razrabotka programmnogo obespecheniya dlya ARM vracha. *Спецтехника и связь*. 2013. N. 4. P. 28–30.

12. **Rapetov A. M., Zuykov A. V., Kusakin I. I.** Obzor algoritmov klassifikatsii izobrazheniy, poluchennukh s pomoshchyu kapsulnogo endoskopa. *Спецтехника и связь*. 2013. N. 4. P. 31–33.

13. **Ayvazyan S. A., Yenyukov I. S., Meshalkin L. D.** *Osnovy modelirovaniya i pervichnaya obrabotka dannykh*. М.: Финансы и статистика, 1983.

УДК 004.27

Е. С. Богодистова, канд. техн. наук, доц., e-mail: alfagreen@yandex.ru,
А. Е. Мамченко, канд. техн. наук., доц., e-mail: mam4enko2011@yandex.ru,
М. И. Шамров, канд. техн. наук, проф., e-mail: shamrovmi@mail.ru,
Московский государственный университет путей сообщения (МИИТ)

Программируемые контроллеры в компьютерных и управляющих системах

Рассматриваются цифровые устройства, относящиеся к классу контроллеров. Выделены группы таких устройств и даются краткие описания реализуемых ими функций в составе компьютерных и управляющих систем.

Ключевые слова: компьютерная система, программируемый логический контроллер, микроконтроллер, системный контроллер, контроллер периферийного устройства

Введение

Термины "контроллер" и "микроконтроллер" вместе со словами, уточняющими их назначение и функции, широко распространены в лексиконе вычислительной техники. Похожий на "контроллер" термин "контролёр" определялся в начале XX века как поверщик, поверитель, браковщик, т. е. как лицо, функцией которого является контроль. В этом смысле он сохраняется и сегодня в отношении контролёров на общественном транспорте и др.

Далее похожий термин "контроллер" используется для электромеханических аппаратов регулирования скорости и режимов работы электропривода в общественном и грузовом транспорте. "Компьютерный" смысл терминов "контроллер" и "микроконтроллер" ближе к своему электромеханическому "собрату", так как означает "управитель", "регулятор", "устройство управления", "управляющее устройство". С английского Controller переводится как "управляющее устройство, регулятор".

В компьютерных и управляющих системах термин "контроллер", как правило, используется в следующих словосочетаниях:

- программируемый логический контроллер (ПЛК);
- однокристалльный микроконтроллер (ОМК) или, иначе, однокристалльная микроЭВМ;
- системные контроллеры, реализующие функциональные подсистемы для обслуживания или расширения возможностей ядра компьютерных систем;
- контроллеры периферийных устройств различного назначения.

Программируемые логические контроллеры

Вместе с развитием персональных компьютеров с 70-х годов прошлого столетия интенсивно разрабатывался, внедрялся и продолжает использоваться класс специализированных управляющих устройств, называемых программируемыми логическими контроллерами, ПЛК (англ. Programmable Logic Controller, PLC) [1]. Их определяют как конфигурируемые и программируемые пользователем системы низового уровня автоматизированных систем управления технологическими процессами (АСУ ТП), которые предназначены для управления локальными и рассредоточенными технологическими процессами и объектами в различных отраслях человеческой жизнедеятельности: промышленности, на транспорте, аграрном секторе и др.

Структуры ПЛК могут быть самыми разнообразными. Однако основной особенностью и отличием ПЛК от традиционного компьютера является то, что основными источниками внешних данных в нем являются модули цифрового и аналогового ввода информации от датчиков объекта, а приемниками результатов обработки — модули цифрового и аналогового вывода на исполнительные механизмы управляемого объекта. Такие модули представляют собой устройства связи с объектом (УСО), составной частью которых являются формирователи сигналов, аналого-цифровые (АЦП) и цифроаналоговые (ЦАП) преобразователи. Другой особенностью ПЛК является функционирование их по фиксированной программе (программе управления объектом), что делает удобным использование

в них процессорного ядра с гарвардской архитектурой [2].

Для программирования ПЛК создавали специальные технологические языки, например язык релейно-контактных схем и др.

ПЛК как универсальная электронная аппаратура отвечает следующим требованиям:

- компактность конструкции, обеспечивающая возможность встраивания их в конструктивы управляемого оборудования;
- модульность, позволяющая определять (иначе, специфицировать) состав средств ПЛК, прежде всего УСО, в соответствии с назначением и функциями управляемого объекта или процесса. С этой целью создается широкий спектр (номенклатура) модулей цифрового и аналогового ввода/вывода;
- обеспечение бесперебойности и качества электропитания, высокой надежности функционирования при управлении ответственными объектами.

Программируемые логические контроллеры обычно создаются в виде линеек устройств различной конфигурации, под которой подразумевается тип и число модулей ввода/вывода, объемы памяти, вычислительные ресурсы и т. п. Кроме того, ПЛК в пределах семейства могут различаться конструктивным исполнением: обычное, пылевлагозащитное, для работы в агрессивных средах и др.

Пионерами-разработчиками ПЛК можно считать фирмы Allen Bradley (США), Coult (Голландия), Siemens (Германия), ВНИИ электромеханики (СССР).

В последние годы наряду с ПЛК широкое распространение получили промышленные компьютеры, для которых характерны большие вычислительные мощности, высокая степень интеллектуализации и взаимодействия с оператором, широкое использование коммуникационных технологий, работа в тяжелых условиях промышленного производства.

Однокристалльные микроконтроллеры

Однокристалльный микроконтроллер (ОМК), или однокристалльная микроЭВМ, является одним из типов микропроцессоров наряду с другими типами — многокристалльными, однокристалльными, многоядерными и другими микропроцессорами. Классический МК содержит, помимо процессора, электрически программируемую память программ, резидентную память данных и аппаратные средства ввода-вывода аналоговых и цифровых сигналов. Такая простая управляющая микросистема на одном кристалле допускает встраивание непосредственно в объект управления. Однокристалльный микроконтроллер является одним из наиболее массовых электронных компонентов и служит основой систем управления техническими объектами в промышленности и на транспорте, используется в бы-

товой аппаратуре, станкостроении, автомобильной промышленности, военном оборудовании, при построении разнообразной компьютерной периферии.

Одно из применений МК — реализация ядра рассмотренных выше программируемых логических контроллеров. Более того, архитектура и свойства МК оказываются весьма приспособленными для реализации функций ПЛК благодаря следующим возможностям:

- использование базового набора команд обработки информации, представленной параллельными кодами разного типа [3];
- адресация битов и наличие расширенной группы логических операций в системе команд ("булевский" процессор), что в соединении с побитовым вводом и выводом делает логическую обработку в микросистемах особенно эффективной;
- развитая подсистема прерываний (большое число запросов, малое время реакции, средства маскирования и управления приоритетами), позволяющая организовать обработку сигналов от датчиков экстренных и аварийных ситуаций;
- аппаратные средства для эффективной реализации подсчета числа событий, интервальных таймеров, генераторов широтно-импульсных сигналов и пр., а также встроенных АЦП и ЦАП;
- последовательные порты ввода-вывода, упрощающие связи с периферией и другими управляемыми системами;
- электрические и конструктивно-технологические решения, предполагающие низкое энергопотребление и возможность управления им, работу в необходимых диапазонах климатических и механических воздействий и т. п.

Примером широко использовавшихся в течение более трех десятилетий МК является семейство MCS-51 (фирма Intel, США), имеющие отечественные аналоги (К1816ВЕ51). В последние годы в мобильных устройствах начинают применяться многоядерные микроконтроллеры.

Системные контроллеры ядра ЭВМ

Управляющий системный контроллер (УСК).

В простейшем случае УСК служит для дешифрации текущего состояния микропроцессора и образования управляющих сигналов для других устройств компьютерной системы и периферийных устройств. Другой функцией УСК является обеспечение требуемой нагрузочной способности формирователей вышеуказанных сигналов. УСК может входить в состав микропроцессорного комплекта ("чипсета"; *англ.* chipset) — набора микросхем, обеспечивающих совместное функционирование базовых устройств компьютера.

В более сложном варианте под системным контроллером понимается северный мост, включающий контроллеры системной шины, шины памяти и шины графического контроллера.

Контроллер динамической памяти (КДП). При хранении информации в динамической памяти (Dynamic Random-Access Memory, DRAM) требуются периодические служебные чтения, называемые регенерацией или рефрешем.

Для реализации рефреша памяти на БИС DRAM в компьютерных системах используют контроллеры динамической памяти (КДП). При обращении к памяти КДП выполняет заданную операцию, формирует групповой адрес и выполняет рефреш части запоминающего массива. В отсутствие обращения контроллер организует фоновую регенерацию ячеек DRAM с помощью внутреннего счетчика адреса.

Так как объем динамической памяти компьютерной системы может быть достаточно большим, а возможности КДП ограничены, рефреш может осуществляться подсистемой из нескольких КДП.

В настоящее время КДП перемещается в микросхемы DRAM, что означает исчезновение КДП как отдельного устройства. Возможно применение динамической памяти без КДП. Примером может служить ПЛК с динамической памятью данных, которая полностью перезагружается информацией о состоянии объекта с периодичностью, меньшей времени хранения чипов DRAM.

Программируемый контроллер прерываний (ПКП). Программируемый контроллер прерываний (Programmable Interrupt Controller, PIC) предназначен для обработки запросов на прерывание процессора посредством реализации следующих функций:

- программная настройка путем загрузки процессором управляющих слов, задающих режимы функционирования, маски, приоритеты и другие параметры;
- прием запросов на прерывание и их запоминание до момента обслуживания;
- выделение запросов в соответствии с их "рангом", определяемым маской и приоритетом;
- образование адреса (или кода для его образования) прерывающей программы и его передача в процессор.

Такой контроллер подключается к процессору и является для него набором программно-доступных управляющих регистров в адресном пространстве ввода-вывода или памяти. На входы контроллера подаются сигналы внешних, в том числе и внутрисистемных, прерываний, а на выходе формируется сигнал прерывания процессора.

ПКП, как и некоторые другие контроллеры, могут входить в состав одной из микросхем чипсета, в частности южного моста.

Контроллер прямого доступа в память (КПДП). Организует один из известных с 60-х годов способ непосредственного обмена данными между периферийным устройством и памятью, минуя процессор (прямой доступ к памяти — ПДП; Direct Memory Access — DMA). КПДП, как и ПКП, программируется процессором путем загрузки параметров области памяти и режима обмена (ввод или вывод).

После настройки КПДП подключается к шинам адреса, данных и управления памяти и выполняет передачу данных между памятью и периферийным устройством. Процессор при этом параллельно может выполнять основную программу.

Контроллеры периферийных устройств

Контроллеры периферийных устройств, коротко "периферийные контроллеры" (Peripheral Controllers), выполняют в общем случае функции сопряжения (интерфейса) ядра компьютерной системы (процессора и памяти) с периферийными устройствами (ПУ), освобождая процессор от непосредственного управления периферией.

Далее рассматриваются разновидности и функции современных периферийных контроллеров компьютерных систем. Несмотря на разнообразие принципов функционирования ПУ и их взаимодействия с ядром, для управления компьютерной периферией используется, как правило, следующая структура: ядро ЭВМ — системный контроллер ПУ — интерфейс обмена — контроллер периферийного аппарата (ПА) — ПА. При этом функции системного контроллера ПУ интегрированы в одну из микросхем чипсета [4].

Контроллеры клавиатуры. Работа клавиатуры обеспечивается двумя контроллерами, связанными через последовательный интерфейс (USB или беспроводный).

Контроллер клавиатуры (КК) встроен непосредственно в клавиатуру и постоянно сканирует матрицу клавиш, выступающую в качестве периферийного аппарата. В процессе сканирования определяется клавиатурный скан-код (номер клавиши) при ее нажатии и отпускании, что позволяет распознать одновременное нажатие нескольких клавиш.

Системный контроллер клавиатуры (СКК) встроен в одну из микросхем чипсета и реализует следующие функции:

- инициализация КК;
- прием данных (клавиатурных скан-кодов нажатия и отпускания клавиш и служебных кодов) от КК по последовательному интерфейсу и их преобразование в параллельный код;
- преобразование клавиатурных скан-кодов в системные скан-коды для настройки на тип клавиатуры;
- формирование запроса прерывания от клавиатуры;
- преобразование управляющей информации, поступившей из ядра ЭВМ, из параллельного кода в последовательный и передача в КК для его настройки.

СКК выполняет определенный набор команд и программируется процессором для задания характеристик клавиатуры и режимов взаимодействия с ядром и КК.

Уровень сложности и "интеллекта" КК и СКК соответствует ОМК.

Контроллеры манипуляторов (указателей) курсора. Работа внешнего устройства позиционирования и оперативного управления (например, "мышь") обеспечивается двумя контроллерами на базе ОМК, соединенными последовательным интерфейсом (USB или беспроводным). Один из них расположен в блоке устройства и обеспечивает управление периферийным аппаратом путем непрерывного сканирования его состояния, отслеживания нажатий клавиш и движения колеса прокрутки. Другой интегрирован, как и системный контроллер клавиатуры, в СБИС чипсета и освобождает центральный процессор от выполнения относительно медленных операций обмена данными, управляющей и осведомительной информацией со встроенным в устройство контроллером.

Если устройство позиционирования встроено в блок клавиатуры, то функции управления им осуществляют контроллеры, общие для двух устройств.

Контроллеры видеоподсистемы. В структуре видеоподсистемы можно выделить два типа контроллеров. Функции системного контроллера реализует в видеоподсистеме *графический контроллер (ГК)*, входящий вместе с видеопамью в состав видеоадаптера. Контроллером периферийного аппарата (видеомонитора) является *дисплейный контроллер (ДК)*. ГК и ДК взаимодействуют через аналоговый (RGB, D-Sub) или цифровой (Digital Visual Interface, DVI) видеointерфейс. ДК получает по этому интерфейсу информацию об интенсивности цветовых (RGB) составляющих элементов изображения (пикселей) и их временном и пространственном расположении по отношению к границам строк и кадров раstra. По аналоговому видеointерфейсу передаются непосредственно RGB-сигналы и сигналы строчной горизонтальной (HSYNC) и кадровой вертикальной (VSYNC) синхронизации, задающие упомянутые границы [5, 6].

Графический контроллер. ГК программируется процессором для задания временных границ строк и кадров раstra, в том числе частоты и других параметров сигналов HSYNC и VSYNC, и разрешения (числа пикселей в строке раstra по горизонтали и числа видимых строк раstra по вертикали).

Основная функция ГК — чтение из видеопамью (кадрового буфера) информации о цвете пикселей и формирование во времени пиксельной и синхронизирующей информации для ДК.

В процессе развития ГК (VGA, SVGA и др.) повышались его интеллектуальность и быстродействие путем внедрения графических ускорителей и графических процессоров (Graphics Processor Units, GPUs) для операций обработки изображений двумерной и трехмерной графики. ГК имеет собственный BIOS на базе флэш-памяти.

Дисплейный контроллер. Классические видеомониторы на базе электронно-лучевой трубки (ЭЛТ)

не имели "интеллекта" и не выполняли никакой предварительной обработки поступающей от ГК информации. Они обеспечивали отображение "на лету" под управлением сигналов аналогового видеointерфейса. Жидкокристаллический дисплей (Liquid Crystal Display, LCD) является интеллектуальным устройством, общее управление которым осуществляет ДК на базе ОМК. Основные функции ДК LCD-дисплея:

- формирование сигналов для управления задней подсветкой слоя жидких кристаллов LCD-панели;
- определение режима работы видеомонитора и типа используемого входного видеointерфейса, обработка синхросигналов HSYNC и VSYNC;
- хранение образа изображения, выводимого на экран, в собственном кадровом буфере монитора;
- масштабирование изображения для подстройки под разрешение LCD-панели;
- формирование опорных напряжений полутонов и сигналов управления строками и столбцами жидкокристаллических ячеек;
- обработка сигналов от лицевой панели управления.

ДК осуществляет хранение данных о настройках монитора и установках пользователя в энергонезависимой перепрограммируемой памяти и использует их при включении и инициализации монитора. С помощью последовательного интерфейса DDX (Data Display Channel), входящего в состав видеointерфейса, проводится автоматическая подстройка монитора и видеоадаптера для работы при наилучших комбинациях разрешения и частоты развертки [7].

Контроллеры магнитных и электронных дисков.

В качестве дисковых накопителей выступают жесткие магнитные диски (НЖМД, Hard Disk Drive, HDD) и их "твердотельные" аналоги на базе микросхем флэш-памяти (Solid State Drive, SSD), называемые электронными дисками. Роль системного контроллера дисковых накопителей выполняет в чипсете контроллер соответствующего стандартного интерфейса: USB для переносных дисков и PATA или SATA (Parallel или Serial AT Attachment) для системных дисков.

Контроллер накопителя на жестком магнитном диске (НЖМД). Контроллер НЖМД как контроллер периферийного аппарата выполняет следующие функции:

- организация кодирования/декодирования и размещения данных на магнитном носителе;
- управление позиционированием блока магнитных головок (БМГ);
- выполнение операций над информационными блоками (запись, чтение, проверка);
- параллельно-последовательное преобразование данных при записи на носитель и обратное — при чтении (воспроизведении) информации с носителя;
- защита от ошибок (помехоустойчивое кодирование, обнаружение и коррекция ошибок в данных);

- организация обмена с ядром и накопителем;
- проверка работоспособности накопителя и собственного оборудования контроллера.

"Интеллект" контроллера представлен тремя элементами: контроллером интерфейса (Hard Disk Controller, HDC), управляющим микропроцессором и цифровым сигнальным процессором (ЦСП; Digital Signal Processor, DSP).

HDC обеспечивает связь с системой (с одной из микросхем чипсета) через параллельный или последовательный интерфейс, а также обслуживает буферную память для промежуточного хранения данных записи-чтения, к которой имеет доступ управляющий микропроцессор.

Микропроцессор управляет всеми компонентами накопителя, включая основной двигатель, осуществляющий вращение пакета дисковых пластин, и поворотный двигатель, выполняющий позиционирование БМГ. Он реализует также функции *сервоконтроллера* для точного позиционирования головок, при этом обеспечивается безопасность диска в случае пропадания питания или остановки основного двигателя.

ЦСП формирует записываемые сигналы и обрабатывает считанные сигналы, обеспечивая высокую плотность записи.

Совершенствование НЖМД и повышение степени интеграции БИС привело к объединению нескольких узлов в одну СБИС HDD-контроллера.

Для электронного диска функции обмена и выполнения операций над информационными блоками аналогичны функциям магнитных дисков (они эмулируются). Отличия в управлении электронным диском связаны с особенностями стирания и записи информации в микросхемах флэш-памяти — электрически репрограммируемых постоянных запоминающих устройствах последних поколений.

Контроллер массива дисков (RAID-контроллер). Одним из вариантов организации работы дисковой системы является RAID-матрица: массив из нескольких дисков с применением избыточности (Redundant Array of Inexpensive (Independent) Disks, RAID). RAID-контроллер (disk array controller, RAID-controller) управляет физическими дисковыми накопителями и представляет их компьютеру как один диск; поддерживает несколько вариантов организации (уровней) RAID-массивов, отличающихся повышенной производительностью и/или надежностью; реализует интерфейс с ядром и интерфейс с управляемыми дисками.

Известно несколько вариантов реализации RAID-контроллеров [8, 9].

Программный RAID-контроллер обеспечивает организацию простейших дисковых массивов на уровне драйверов операционной системы. Характеризуется низкой производительностью и снижает производительность системы в целом.

Интегрированный RAID-контроллер размещается внутри компьютера как карта расширения либо

встраивается в системную плату как часть чипсета. Он обслуживает несложный дисковый массив из нескольких PATA и SATA жестких дисков. Поддерживает, как правило, базовые уровни RAID и обеспечивает высокую скорость работы. Является программно-аппаратным, не имеет собственного процессора и использует вычислительные мощности центрального процессора и системной памяти. Системная плата может включать несколько интегрированных RAID-контроллеров, например, в южный мост чипсета.

Аппаратный RAID-контроллер выполняется в виде отдельной платы в составе системного блока либо в виде специальной стойки, содержащей массив накопителей. Обеспечивает минимальные задержки. Включает:

- собственный процессор и буферную память для координации дисковых массивов;
- собственную кэш-память на плате, при этом возможна работа от аккумулятора для сохранения содержимого кэш-памяти в случае сбоя электропитания;
- средства аппаратной коррекции ошибок;
- собственный BIOS, обычно на базе флэш-памяти. Позволяет загружаться с RAID-массива. Поддерживает различные способы использования дисков:
- гибридные массивы дисков разных типов и уровней RAID;
- простой массив независимых дисков — использование дисков по одному, без организации RAID массива (режим Just a Bunch Of Disks, JBOD);
- коммутируемые жесткие диски (Switched Bunch Of Disks, SBOD), представляющие собой группу жестких дисков (JBOD) со встроенной системой коммутации;
- диски для горячей замены, резервные диски.

Контроллер накопителя на оптических дисках.

Роль системного контроллера накопителя на оптических дисках выполняет в чипсете контроллер ответственного стандартного интерфейса: IDE (PATA) или SCSI (Small Computer System Interface). "Интеллект" накопителя представлен однокристалльным микроконтроллером (одним-двумя) и ЦСП, в функции которых входит:

- поддержка обмена с ядром;
- управление загрузочным устройством;
- координация систем автоматического слежения за поверхностью диска и спиральной дорожкой с данными и управления исполнительными элементами и узлами накопителя;
- синхронизация процессов записи-чтения, формирование сигналов записи и декодирование считанных сигналов;
- защита от ошибок.

Встроенные системы автоматического регулирования и управления накопителя обеспечивают:

- регулирование расстояния между диском и линзой объектива ("автофокус");
- радиальную подачу считывающего/записывающего блока путем управления серводвигателем

перемещения оптической головки для захвата очередной группы витков спиральной дорожки в поле зрения линзы;

- слежение за спиральной дорожкой для предотвращения сбоев в позиционировании луча лазера вследствие радиальных биений диска (автотрекинг);
- в режиме CLV (Constant Linear Velocity) — постоянную линейную скорость перемещения спиральной дорожки относительно считывающего/записывающего блока; эта сервосистема управляет двигателем вращения диска;
- неизменную мощность излучения лазера в заданном режиме независимо от температуры окружающей среды и перепадов напряжения.

Контроллер принтера. Роль системного контроллера принтера выполняет в чипсете контроллер соответствующего стандартного интерфейса: параллельного (Line Printer, LPT) или USB. "Интеллект" растрового принтера представляют однокристалльный микроконтроллер для двунаправленного обмена данными с системой и микропроцессор, обеспечивающий печать при взаимодействии с датчиками и исполнительными элементами принтера.

Формирование растровой карты изображения (растеризация, или растривание) в соответствии с выбранным разрешением принтера требует вычислительных ресурсов и памяти (оперативной и/или дисковой). Если растеризация выполняется компьютером, то буфер памяти самого принтера рассчитан на одну строку изображения (один проход печатающей головки), а переполнению буфера принтера препятствует приостановка процесса передачи данных по интерфейсу. В случае, когда драйвер принтера сжимает растеризованный образ, контроллер принтера выполняет декомпрессию.

При формировании растра непосредственно принтером передаваемые ему данные представляют собой информацию о графических объектах в формате языка описания страниц (Page Description Language, PDL). При этом поступающие в принтер данные обрабатываются растровым процессором (Raster Image Processor, RIP), который формирует в ОЗУ принтера растровое изображение целой страницы для последующего переноса на бумагу.

USB-контроллер. Обеспечивает двухсторонний обмен данными в режиме прямого доступа между ядром компьютера и периферийными устройствами, подключенными к универсальной последовательной шине USB.

Двухсторонний обмен требует реализации определенных функций USB как периферийным устройством, так и системой (соответственно функции работы Device и Host). Выпускаются микросхемы контроллеров USB, поддерживающих функции только периферийного устройства (функция работы Device), микросхемы хост-контроллеров концентраторов (функция Host) и такие, которые могут быть настроены на выполнение любой из двух

функций. Хост-контроллер шины USB может быть интегрирован в одну из микросхем чипсета либо располагаться в виде дискретного чипа на системной плате или плате расширения.

Микросхемы чипсета как контроллеры

Микросхемы чипсета также называют контроллерами. *Контроллер-концентратор памяти* (северный мост) содержит, в свою очередь, контроллеры системной шины, шины памяти и шины графического контроллера. *Контроллер-концентратор ввода-вывода* (южный мост) содержит контроллеры интерфейсов накопителей, устройств ввода-вывода и шин для подключения устройств мультимедиа и периферии. Третьей может быть микросхема *мультиконтроллера*, которая реализует функции контроллеров "устаревших" низкоскоростных интерфейсов передачи данных: последовательного (RS-232) и параллельного (LPT) интерфейсов, контроллеров манипулятора "мышь" и клавиатуры, контроллера накопителя на гибком магнитном диске. К этой микросхеме подключаются также устройства хранения криптографической информации.

Иногда основные микросхемы объединяются в один чип, называемый *системным контроллером-концентратором* (System Controller Hub, SCH).

Многообразие цифровых устройств, относящихся к классу контроллеров, не ограничивается рассмотренными. Назовем некоторые другие типы контроллеров, не упомянутые выше.

Контроллеры сенсорных экранов совмещают функции отображения информации с функциями устройств ввода и оперативного управления и имеют уровень "интеллекта", соответствующий ОМК.

Кэш-контроллеры (Cache Controllers) управляют записью, хранением и поиском часто используемых данных и команд в кэш-памяти, обеспечивая ускорение доступа к основной оперативной (со стороны процессора) или дисковой памяти (со стороны ядра и дискового контроллера).

В чипсет или процессор часто интегрированы:

- *звуковой контроллер* (звуковая плата, Sound Card), дополненный аудиокодеком, позволяющий выводить звук на акустические системы и/или записывать его;
- *сетевой контроллер* (сетевая карта, сетевой адаптер, Network Interface Controller, NIC), обеспечивающий взаимодействие компьютера с другими устройствами сети;
- *SATA-контроллер* дискового накопителя, о котором уже шла речь выше;
- *SCSI-контроллер* обеспечивает работу высокопроизводительной шины SCSI; устанавливается в слот расширения или интегрирован в материнскую плату.

Универсальный интерфейсный контроллер представляет собой плату, объединяющую несколько различных интерфейсов и используемую при огра-

ниченном числе свободных слотов расширения на материнской плате. Простые контроллеры этого рода позволяют добавить в систему максимальное число интерфейсных портов одного типа. Гибридные, комбинированные контроллеры объединяют различные интерфейсы (например, USB и FireWire, SATA и PATA).

Часто применяют термин "контроллер шины", имея в виду контроллер системного или периферийного интерфейса.

Заключение

Предложенный обзор устройств, называемых контроллерами, не претендует на всеохватность и тем не менее демонстрирует многообразие их применений в составе современных вычислительных систем. Большинство таких контроллеров, будучи программными автоматами, обладают "интеллектом" и взаимодействуют друг с другом с использованием различных интерфейсов и элементов сетевых технологий. Это позволяет представлять современную компьютерную систему в виде множества узлов, объединенных в сетеподобную структуру с высо-

кой степенью распараллеливания вычислительного процесса.

Список литературы

1. Мишель Ж. Программируемые контроллеры: архитектура и применение: Пер. с фр. М.: Машиностроение, 1992. 320 с.
2. Мамченко А. Е. Образовательно-методологический аспект "принстонской" и "гарвардской" архитектур процессоров вычислительных систем // Информационные технологии. 2009. № 7. С. 13—18.
3. Мамченко А. Е., Першеев В. Г. О кодах представления чисел с фиксированной точкой (запятой) в компьютерах и вычислительных системах // Информационные технологии. 2011. № 4. С. 51—59.
4. Колесниченко О. В., Шишигин И. В., Соломенчук В. Г. Аппаратные средства РС. СПб.: БХВ-Петербург, 2010. 800 с.
5. Попов С. Н. Аппаратные средства мультимедиа. Видео-система РС / Под ред. О. В. Колесниченко, И. В. Шишигина. СПб.: БХВ-Петербург, Арлит, 2000. 400 с.
6. Архипов О. П., Зыкова З. П. Коррекция детализации представлений RGB-изображений на периферийных устройствах ПЭВМ // Информационные технологии. 2013. № 2. С. 56—60.
7. URL: <http://www.mirpu.ru/lcd/75-shtmatrenics/163-tftlg1.html> Схемотехника TFT мониторов.
8. Егоров Г. А., Шяудкулис В. И., Финотти М. Проблемы построения современных архивных хранилищ данных // Информационные технологии. 2012. № 12. С. 8—18.
9. URL: <http://www.ixbt.com/storage/raids.html> RAID Levels

E. S. Bogodistova, Assistant Professor, e-mail: alfagreen@yandex.ru,

A. E. Mamchenko, Assistant Professor, e-mail: mam4enko2011@yandex.ru,

M. I. Schamrov, Professor, e-mail: shamrovmi@mail.ru,

Moscow State University of Railway Engineering (MIIT)

Programmable Controllers in Computer and Control Systems

Digital devices related to controller class are discussed. Groups of such devices are determined as architecture elements of computer and control systems: single-chip microcontroller, programmable logic controller, computer system controller and controllers of peripheral devices. Features of the organization of the single-chip microcontrollers used as base elements of many other controller types are considered. Design principles of programmable logic controller and requirements to such a device are formulated. Functions of the system controllers as a computer kernel providing its adjustment and functioning are listed. The two-level management of a peripheral subsystem is analyzed and controllers of the basic peripheral units are considered for keyboards, pointing devices, a video subsystem, the printer, stores on magnetic, optical and electronic disks, disk arrays. USB-controller functions as universal connection element for peripherals are listed. Use of single-chip microcontrollers in a kernel and periphery of computer is shown. The role of a microprocessor chipset as a set of controllers with different functions is discussed. It is shown that the modern computer system includes a great number of the various controllers allowing to parallelize and to accelerate computing process.

Keywords: computer system, control system, single-chip microcontroller, programmable logic controller, system controller, peripheral device controller, keyboard, positioning device, video subsystem, printer, magnetic, optical and electronic disks, disk array

References

1. Mishel' Zh. Programmiruemye kontrollery: arhitektura i primeneniye. Per. s fr. M.: Mashinostroeniye, 1992. 320 p.
2. Mamchenko A. E. Obrazovatel'no-metodologicheskij aspekt "prinostonskoj" i "garvardskoj" arhitektur processorov vychislitel'nyh sistem. *Informacionnyye tehnologii*. 2009. N. 7. P. 13—18.
3. Mamchenko A. E., Persheev V. G. O kodah predstavleniya chisel s fiksirovannoj tochkoj (zapjatoj) v kompjuterah i vychislitel'nyh sistemah. *Informacionnyye tehnologii*. 2011. N. 4. P. 51—59.
4. Kolesnichenko O. V., Shishigin I. V., Solomenchuk V. G. Apparatsnyye sredstva PC. 6-e izd., pererab. i dop. SPb.: BHV-Peterburg, 2010. 800 p.
5. Popov S. N. Apparatsnyye sredstva mul'timedia. Videosistema PC / Pod red. O. V. Kolesnichenko, I. V. Shishigina. SPb.: BHV-Peterburg, Arlit, 2000. 400 p.
6. Arhipov O. P., Zyкова Z. P. Korrekciya detalizacii predstavlenij RGB-izobrazhenij na periferijnyh ustrojstvah PJeVM. *Informacionnyye tehnologii*. 2013. N. 2. P. 56—60.
7. URL: <http://www.mirpu.ru/lcd/75-shtmatrenics/163-tftlg1.html> Shemotehnika TFT monitorov
8. Egorov G. A., Shjaukulis V. I., Finotti M. Problemy postroenja sovremennyh arhivnyh hranilishh dannyh. *Informacionnyye tehnologii*. 2012. N. 12. P. 8—18.
9. URL: <http://www.ixbt.com/storage/raids.html> RAID Levels

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ CRYPTOSAFETY INFORMATION

УДК 004.421.2

М. Р. Богданов, канд. биол. наук, доц., e-mail: bogdanov_marat@mail.ru,
А. В. Захаров, канд. физ.-мат. наук, доц., e-mail: ralzax@yandex.ru,
Ю. З. Габидуллин, преподаватель,
А. А. Думчиков, студент, e-mail: redfoxufa@gmail.com,
В. Г. Дмитриев, канд. пед. наук, доц., e-mail: dmitriyeva@yandex.ru,
Башкирский государственный педагогический университет им. М. Акмуллы, г. Уфа

Кодирование текста с помощью музыки

Работа посвящена методу кодирования текстовых сообщений с помощью музыки. Для кодирования используется технология MusicXML. Декодирование осуществляется методом вейвлет-анализа. Предлагаемый подход обладает высокой криптографической стойкостью.

Ключевые слова: криптография, MusicXML, вейвлет-анализ

Разоблачения мистера Сноудена обнажили очень интересную проблему. Для нормальной работы современного бизнеса, общественных институтов, правительственных организаций и частных лиц требуется доверие. Доверие обеспечивается алгоритмами шифрования, встроенными в браузеры, операционные системы, веб-приложения и т. д. Было бы странно, если бы спецслужбы не занимались взломом шифров в целях получения доступа к конфиденциальной информации. По некоторым данным, АНБ уделяет пристальное внимание взлому таких протоколов, как SSL, VPN и 4G [1]. Если взлом не удастся осуществить простым перебором ключей, как например, в случае 256-битного шифрования, то алгоритм шифрования ослабляется с помощью уязвимостей. Имеются сведения, что таким образом АНБ ослабила алгоритм шифрования RSA [2]. По данным газеты Гардиан, АНБ имеет доступ к Интернет-трафику большой четверки — Hotmail, Google, Yahoo и Facebook [3].

Намеренное ослабление изначально стойких алгоритмов шифрования правительственными спецслужбами ставит законный вопрос: не смогут ли этими уязвимостями воспользоваться хакеры? Второй вопрос: как обеспечить конфиденциальность пересылаемых данных. Решение видится в использовании нетрадиционных малораспространенных алгоритмов шифрования.

Шифрование с помощью музыки

В Калифорнийском университете в Лос-Анджелесе на протяжении ряда лет ведется проект Gene2Music, целью которого является генерация музыкальных произведений на основе первичной

последовательности белков и ДНК. В качестве прототипа был использован фермент тимидилат-синтаза А человека (ThyA) [4]. Участники проекта обычно использовали ноты для кодирования аминокислот или нуклеотидов.

В январе 2005 г. вышла первая версия формата MusicXML, разработанного компанией Recordare LLC. Этот формат, основанный на XML, описывает западную музыкальную нотацию. Он разработан для обмена партитурами, созданными в разных музыкальных редакторах. В августе 2011 г. вышла третья версия стандарта с поддержкой виртуальных инструментов в версиях как DTD (Document Type Definition), так и XSD (XML Schema Definition). Формат MusicXML поддерживается многими редакторами партитур, включая Finale, Sibelius, and MuseScore, большинством программ оптического распознавания музыкальных символов, включая SmartScore и PhotoScore, многими музыкальными секвенсорами, включая Cubase, Logic Pro, SONAR и Rosegarden. Кроме того, благодаря возможностям HTML5 можно воспроизводить высококачественную музыку, записанную в MusicXML-формате в веб-браузере [5].

Формат MusicXML очень скрупулезно описывает музыкальное произведение, объем тестового файла XML-документа сопоставим по объему с музыкальным wav-файлом. Парсеры MusicXML, такие как MuseScore, генерируют 16-битный стереозвук с частотой дискретизации 44 100 Гц. Музыкальное произведение, записанное в формате MusicXML, состоит из небольших фрагментов — аккордов, длительностью примерно 2...4 с. Идея шифрования заключается в связывании того или иного аккорда с определенным символом. На практике идея ре-

Таблица 1

Образец кодовой таблицы

№	Сим-вол	№	Сим-вол	№	Сим-вол	№	Сим-вол	№	Сим-вол
1	A	15	O	29	c	43	q	57	4
2	B	16	P	30	d	44	r	58	5
3	C	17	Q	31	e	45	s	59	6
4	D	18	R	32	f	46	t	60	7
5	E	19	S	33	g	47	u	61	8
6	F	20	T	34	h	48	v	62	9
7	G	21	U	35	i	49	w	63	Пробел
8	H	22	V	36	j	50	x	64	.
9	I	23	W	37	k	51	y	65	,
10	J	24	X	38	l	52	z	66	(
11	K	25	Y	39	m	53	0	67)
12	L	26	Z	40	n	54	1	68	!
13	M	27	a	41	o	55	2		
14	N	28	b	42	p	56	3		

лизуется следующим образом. Берется достаточно продолжительная эталонная мелодия, например фортепианный или скрипичный концерт, и кодируется в MusicXML-формат. В работе мы использовали симфонию, состоящую из 68 аккордов. Было задействовано 14 музыкальных инструментов. Объем XML-документа составил 1,4 Мбайт. Далее была составлена кодовая таблица (она может быть произвольной) (табл. 1).

Предположим, что мы хотим закодировать фразу: "Hello, World!". В этом случае будет сгенерировано музыкальное произведение, образованное аккордами с номерами: 8, 31, 38, 38, 41, 65, 63, 23, 41, 44, 38, 30, 68.

Зашифрованное сообщение можно передавать по открытым каналам передачи данных или в виде XML-документов, или в виде звуковых файлов, например в wav-формате.

Декодирование зашифрованного сообщения

Расшифровку скрытого сообщения, записанного в формате MusicXML, можно осуществлять с помощью парсера, содержащего кодовую таблицу.

Для декодирования сообщения, скрытого в звуковом файле, можно использовать вейвлет-анализ. Вейвлет-анализ нам кажется более перспективным. Вейвлет — это некая функция, заданная таблично. Она позволяет определенным образом характеризовать сигнал. Вейвлет-анализ широко используется при анализе сейсмических, звуковых и биомедицинских сигналов. Если немного упростить ситуацию, то распознавание выполняется следующим образом. Вначале известный сигнал подвергается процедуре детектирования. Детектирование позволяет избавиться от избыточности и выявить характерные особенности сигнала. Затем выбирается характерная область детектированного сигнала. Эта область подвергается нормализации. Данная процедура позволяет компенсировать влияние амплитуды на распознавание. Часть детектированного сигнала, подвергнутого нормализации, является вейвлетом. Вейвлеты сохраняются для дальнейшего использования. В процессе распознавания неизвестный сигнал сравнивается с известным вейвлетом. В идеале вейвлет должен давать высокую корреляцию с целевым сигналом и низкую корреляцию с другими сигналами.

Как уже говорилось выше, такие парсеры MusicXML-документов, как MuseScore, генерируют 16-битный стереозвук с частотой дискретизации 44 100 Гц. Средняя продолжительность аккорда составляет 3 с, т. е. один символ сообщения кодируется массивом из 264 600 элементов (3 с, 2 канала, 44 100 Гц). На первом этапе декодирования сообщения мы избавлялись от избыточности звукового файла, преобразуя его в монофонический 8-битный сигнал с частотой дискретизации 5512 Гц. Такой подход используют многие Интернет-компании, такие как

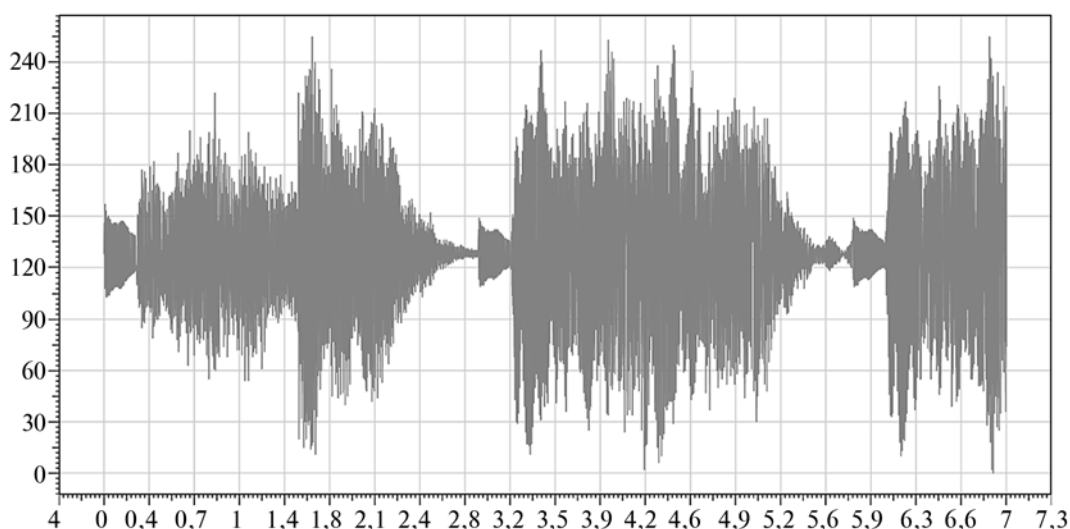


Рис. 1. Спектрограмма звукового файла (8 бит, 5512 Гц, моно). По шкале абсцисс отложено время

Google, при анализе музыкального контента на предмет нарушения авторских прав (аудиофингерпринтирование звука) [6]. В итоге мы получили массив из 16 536 элементов на символ (3 с, 5512 Гц) (рис. 1).

Создание вейвлетов

Детектирование проводится следующим образом. Вначале выбирается рамка детектирования $Frame$. В нашем случае рамка содержит 100 значений (этот параметр подбирается экспериментально). Далее весь числовой массив A , состоящий из N значений, делится на $N/Frame$ участков. Внутри каждой рамки детектирования находим сумму квадратов

$$\text{sum}_j = \sum_{i=1}^{Frame} A_i^2,$$

где j — номер рамки; i — номер элемента в рамке [7].

Далее мы получаем массив детектированного сигнала B , где элементы массива B_j вычисляются по формуле:

$$B_j = \sqrt{\frac{\text{sum}_j}{Frame}}.$$

На рис. 2 показан детектированный сигнал.

Проведя детектирование сигнала, мы выбираем область, характерную для конкретного сигнала. При этом необходимо выбрать начальный B_{\min} и конечный B_{\max} участки детектированного сигнала. В нашем случае $B_{\min} = 1$, а $B_{\max} = 121$, т. е. длина вейвлета составила $121 - 1 = 120$ значений (эти параметры подбираются экспериментально). Далее на выбранном участке детектированного сигнала (с 1-го по 121-й) проводится процедура нормализации. Для этого находят:

- среднее арифметическое

$$r = \frac{1}{m} \sum_{B_{\min}}^{B_{\max}} B_j,$$

где $m = B_{\max} - B_{\min}$;

- сумма квадратов отклонений

$$\text{sum}_{sqr} = \sum_{B_{\min}}^{B_{\max}} (B_j - r)^2,$$

- дисперсия

$$\sigma = \sqrt{\frac{\text{sum}_{sqr}}{m}}.$$

Зная r и σ , можно найти вейвлет, являющийся массивом $V_j = \frac{B_j - r}{\sigma}$ (рис. 3).

Распознавание

На вход системы подается неизвестный сигнал U_i . Далее проводится детектирование сигнала с такой же рамкой $Frame$ (в нашем случае 100 отсчетов), что и при создании вейвлета. Получаем массив детектированного сигнала C_j (в 100 раз меньше отсчетов, чем в исходном сигнале). Затем выбирается шаг распознавания $Step$, равный размеру вейвлета m (в нашем случае $m = 120$). Внутри шага распознавания определяется коэффициент корреляции участка исследуемого сигнала C_j , равного $Step$, со сравнимым вейвлетом. Определение коэффициента корреляции COR_j проводили следующим образом. На исследуемом участке находили среднее арифметическое r и дисперсию σ способом, используемым при создании

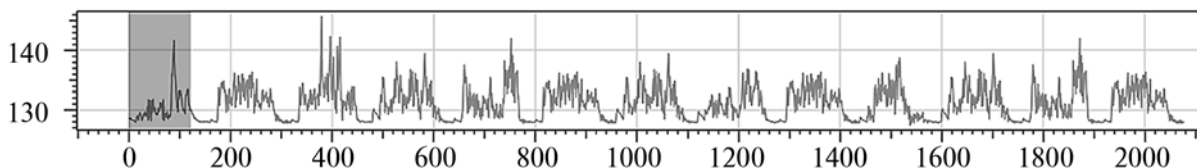


Рис. 2. Детектированный сигнал звукового файла. По оси абсцисс отложены номера элементов массива, по оси ординат — амплитуда сигнала. На рисунке видны музыкальные аккорды

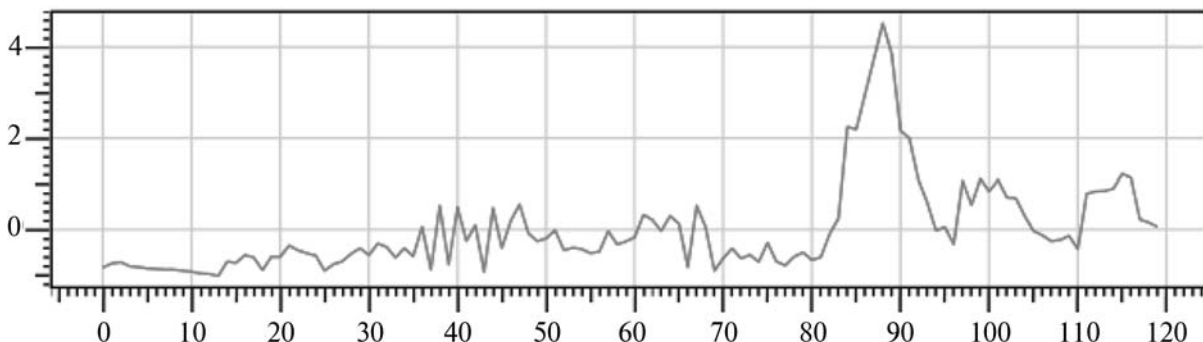


Рис. 3. Вейвлет музыкального аккорда. По оси абсцисс отложены номера элементов массива, по оси ординат — амплитуда сигнала

вейвлета. На основании r и σ нормировали выбранный участок исследуемого детектированного сигнала, получая массив D_j . Имея нормированный детектированный сигнал D_j и вейвлет V_j , находим коэффициент корреляции (рис. 4)

$$\text{COR}_j = \sqrt{\frac{\sum_j^{j+\text{Step}} D_j V_j}{\text{Step}}}$$

Результаты и обсуждение

Нами было разработано программное обеспечение, предназначенное для кодирования и декодирования текстов в музыкальные файлы и из музыкальных файлов. Коэффициент корреляции вейвлетов с целевыми сигналами был равен единице, корреляция к другим сигналам варьировалась в пределах от 0,5946 до 0,9811 (табл. 2).

Если задать критерий распознавания: корреляция к целевому сигналу $> 0,999999$, то распознавание будет 100 %-ным.

Криптографическую стойкость предлагаемого подхода можно повысить за счет:

- избыточности (можно сделать так, чтобы не все аккорды были кодируемыми. Это затруднит частотный криптоанализ);
- использования перекрывающихся участков музыкальных аккордов для кодирования информации;
- внесения незначительных изменений (2...3 замены) в фрагмент звукового файла, кодирующего символ, что приведет к изменению коэффициента корреляции примерно на 0,001. Этого оказывается достаточно для уверенного распознавания символов. То есть разные символы можно кодировать одним и тем же музыкальным аккордом, в котором на 16 536 элементов встречаются 2...3 замены.

Список литературы

1. **Revealed:** The NSA's Secret Campaign to Crack, Undermine Internet Security. URL: <http://www.propublica.org/article/the-nsas-secret-campaign-to-crack-undermine-internet-encryption>
2. **Major security firm: Avoid NSA-linked algorithm (FRIDAY, SEP 20, 2013 06:47 PM EST).** URL: http://www.salon.com/2013/09/20/major_security_firm_avoid_nsa_linked_algorithm/
3. **Revelations** Around NSA, GCHQ Operations to Crack Encryption & the Need for More Whistleblowers. By: Kevin Gosztoła Friday

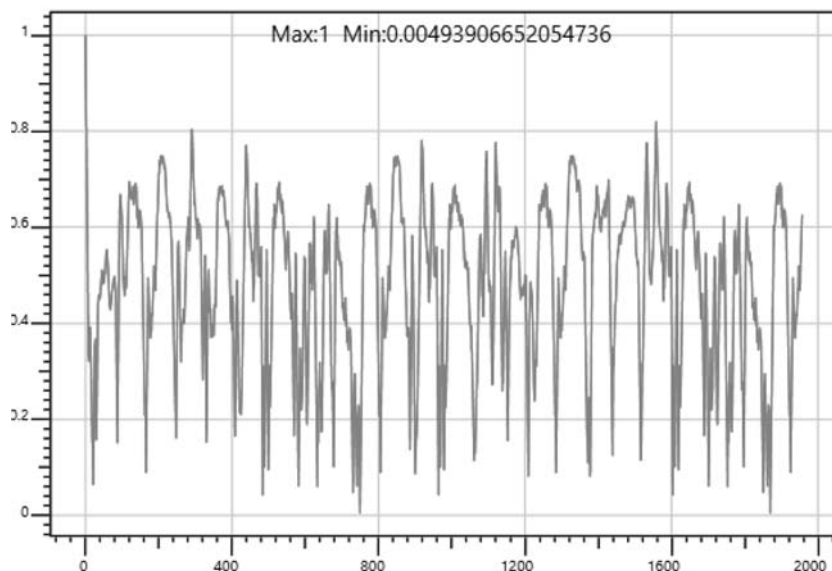


Рис. 4. Корреляционная функция. По оси абсцисс отложены номера элементов массива, по оси ординат — коэффициент корреляции

Таблица 2

Фрагмент корреляционной матрицы вейвлетов и музыкальных аккордов

Вейвлет/ Сигнал	1	2	3	4	5	6	...	67	68
1	1	0,822	0,774	0,791	0,733	0,704	...	0,737	0,738
2	0,832	1	0,662	0,767	0,781	0,74	...	0,771	0,68
3	0,782	0,663	1	0,687	0,697	0,719	...	0,65	0,688
4	0,796	0,776	0,685	1	0,868	0,781	...	0,825	0,788
5	0,741	0,781	0,709	0,868	1	0,795	...	0,834	0,771
6	0,715	0,727	0,728	0,75	0,763	1	...	0,817	0,775
7	0,735	0,712	0,692	0,731	0,762	0,802	...	0,774	0,763
8	0,733	0,747	0,735	0,764	0,785	0,967	...	0,815	0,776
9	0,698	0,734	0,866	0,741	0,747	0,809	...	0,728	0,716
...
66	0,731	0,717	0,758	0,844	0,862	0,822	...	0,8	0,784
67	0,762	0,714	0,65	0,689	0,77	0,764	...	1	0,83
68	0,719	0,758	0,644	0,814	0,816	0,806	...	0,849	1

September 6, 2013 11:13 am. URL: <http://dissenter.firedoglake.com/2013/09/06/revelations-around-nsa-gchq-operations-to-crack-encryption-the-need-for-more-whistleblowers/>

4. **Gene** to Music site. URL: http://www.mimg.ucla.edu/faculty/miller_jh/gene2music/previouswork.html

5. **Руководство** пользователя стандарта MusicXML. URL: <http://www.musicxml.com/wp-content/uploads/2012/12/musicxml-tutorial.pdf>

6. **Shumeet Baluja, Michele Covell.** Audio fingerprinting: combining computer vision & data stream processing. URL: <http://static.googleusercontent.com/media/research.google.com/ru//pubs/archive/33031.pdf>

7. **Bouden Toufik and Nibouche Mokhtar.** The Wavelet Transform for Image Processing Applications. 2012.

8. **Богданов М. Р., Думчиков А. А.** Способ изучения разнообразия орнитофауны методом вейвлет-анализа // Известия Самарского научного центра РАН. 2013. № 3 (4). С. 1232—1236.

9. **Богданов М. Р., Думчиков А. А.** Метод распознавания электрокардиограмм методом вейвлет-анализа. Свидетельство о государственной регистрации программы для ЭВМ № 2013661588.22.10.2013.

M. R. Bogdanov, Associate Professor, e-mail: bogdanov_marat@mail.ru, **A. V. Zakharov**, Associate Professor, **Ju. Z. Gabidullin**, Lecturer, **A. A. Dumchikov**, Student, **V. G. Dmitriev**, Associate Professor, M. Akmullah named after Bashkir State Pedagogical University

Coding Text with Music

Confidence is very important for business, social institutions, government and privacy. A number of cryptographical algorithms are using for achieving the confidence. Unfortunately, some telecommunication technologies including SSL, VPN, 4G and others have vulnerabilities. Using of non-traditional and rare crypto technologies could achieve the confidence. We offer an approach for coding text to music using MusicXML technology. MusicXML protocol describes music very thoroughly. Size of xml-document generated is comparable to mp3 version of the sound. There are MusicXML parsers generating high quality sound files based on the xml-documents (for example, MuseScore). A musical composition according to MusicXML notation consists from logical units of music named chords. We can code the original message into the MusicXML document using a simple coding table. Then you can send an encrypted message to receiver, or in the form of xml-document or in the generated audio file. The code is very redundancy and hard to crypto analysis. We offer decode the initial message with wavelet-analysis. There are a number of ways for strengthen of cryptographic security.

Keywords: cryptography, MusicXML, wavelet-analysis

References

1. **Revealed:** The NSA's Secret Campaign to Crack, Undermine Internet Security. URL: <http://www.propublica.org/article/the-nsas-secret-campaign-to-crack-undermine-internet-encryption>
2. **Major** security firm: Avoid NSA-linked algorithm (FRIDAY, SEP 20, 2013 06:47 PM EST). URL: http://www.salon.com/2013/09/20/maior_security_firm_avoid_nsa_linked_algorithm/
3. **Revelations** Around NSA, GCHQ Operations to Crack Encryption & the Need for More Whistleblowers. By: Kevin Gosztola Friday September 6, 2013 11:13 am. URL: <http://dissenter.firedoglake.com/2013/09/06/revelations-around-nsa-gchq-operations-to-crack-encrvption-the-need-for-more-whistleblowers/>
4. **Gene** to Music site. URL: http://www.mimg.ucla.edu/faculty/miller_jh/gene2music/previouswork.html
5. **MusicXML** tutorial. URL: <http://www.musicxml.com/wp-content/uploads/2012/12/musicxml-tutorial.pdf>
6. **Shumeet Baluja, Michele Covell.** Audio fingerprinting: combining computer vision & data stream processing. URL: <http://static.googleusercontent.com/media/research.google.com/ru/pubs/archive/33031.pdf>
7. **Bouden Toufik and Nibouche Mokhtar.** The Wavelet Transform for Image Processing Applications. 2012.
8. **Bogdanov M. R., Dumchikov A. A.** Sposob izuchenia raznoobrazia ornitophauny metodom veivlet-analiza. *Izvestia Samarskogo nauchnogo centra RAN.* 2013. N. 3 (4). P. 1232—1236.
9. **Bogdanov M. R., Dumchikov A. A.** Metod raspoanavania electrocardiogram metodom veivlet-analiza. Svidetelstvo o gosudarstvennoi registracii program dlya EVM # 2013661588. 22.10.2013.

ЖУРНАЛ В ЖУРНАЛЕ



**НЕЙРОСЕТЕВЫЕ
ТЕХНОЛОГИИ**

№ 11
НОЯБРЬ
2014

Главный редактор:

ГАЛУШКИН А. И.

Редакционная коллегия:

АВЕДЬЯН Э. Д.
БАЗИЯН Б. Х.
БЕНЕВОЛЕНСКИЙ С. Б.
БОРИСОВ В. В.
ГОРБАЧЕНКО В. И.
ЖДАНОВ А. А.
ЗЕФИРОВ Н. С.
ЗОЗУЛЯ Ю. И.
КРИЖИЖАНОВСКИЙ Б. В.
КУДРЯВЦЕВ В. Б.
КУЛИК С. Д.
КУРАВСКИЙ Л. С.
РЕДЬКО В. Г.
РУДИНСКИЙ А. В.
СИМОРОВ С. Н.
ФЕДУЛОВ А. С.
ЧЕРВЯКОВ Н. И.

**Иностранные
члены редколлегии:**

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б. М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Барский А. Б.

Рейтинговые системы на логических нейронных сетях 66

Аникин В. И., Карманова А. А.

Обучение искусственной нейронной сети Кохонена клеточным автоматом 73

А. Б. Барский, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,
Московский государственный университет путей сообщения (МИИТ)

Рейтинговые системы на логических нейронных сетях

Исследуется возможность применения логических нейронных сетей в системах определения рейтинга объекта по его основным показателям, найденным с учетом неопределенности. Технология применима для банковского мониторинга, мониторинга в сфере образования, для определения риска — странового, страхового, военно-стратегического, в управлении надежностью сложных систем и пр. Используются механизмы "бесформульных" ассоциативных вычислений по базе знаний, накопленной на основе опытных данных. Исследования ориентированы на разработку унифицированного программного приложения для широкого круга "рейтинговых" задач.

Ключевые слова: логическая нейронная сеть, функция активации, матрица следования, рейтинговая система, мониторинг

Введение

Логические нейронные сети [1—8] отличаются от традиционных ("классических" — в свете современной теории нейронных сетей) тем, что их переменными являются не числовые значения аргументов (денежные суммы, грузоподъемность, число исполнителей и т. д.), а *достоверность высказываний о принадлежности данных*. Например, указанная достоверность может являться предполагаемой вероятностью того, что снег на тротуаре белый, большой страдает почечной недостаточностью, грузоподъемность транспортного средства составляет три тонны, скрываемый капитал олигарха оценивается в интервале \$30...35 млрд, след НЛО образует правильный треугольник и т. д. Здесь уже видна возможность объединения в единой системе логического вывода, на основе причинно-следственных связей, информации самой разной, казалось бы, несовместимой природы, назначения, формы представления и типов данных! Ведь все данные оказываются представленными в единой системе измерения величинами сигналов из диапазона от нуля до условной единицы, обусловленной техническим средством воплощения. При таком представлении возможны арифметические действия над данными в соответствии с *пороговой функцией активации нейрона* и голосование в пользу предпочтительного решения. Разве это не основная идея работы мозга?

В основе теории логических нейронных сетей лежит *математическая логика событий*, использующая самые простые, начальные понятия математической логики, а точнее — булевой алгебры, и теории вероятностей.

Булевы переменные, принимающие значения 0 и 1, обобщаются на весь диапазон от нуля до условной единицы, превращаясь в действительные. Это обобщение и соответствует переходу к рассмотрению достоверности высказываний о наступлении события вместо строгого суждения о том, что данное высказывание либо истинно, либо ложно. Исполь-

зование исчерпывающих множеств событий в пределах одного фактора приводит к тому, что отрицание каждого события превращается в дизъюнкцию остальных событий данного фактора ($\text{не(лето)} = \langle \text{весна} \rangle \vee \langle \text{осень} \rangle \vee \langle \text{зима} \rangle$). Тогда совершенная дизъюнктивная нормальная форма каждой логической функции преобразуется в дизъюнкцию конъюнкций. Способом "размножения решений" логическая нейронная сеть превращается в однослойную. Это определяет общий вид логических нейронных сетей в системах принятия решений (СПР) и достаточность их рассмотрения.

Таким образом, переход к достоверности высказываний о принадлежности данных обеспечил прорыв из исключительно числовой области применений, характерной для "классических" нейронных сетей, к практически неограниченному диапазону возможных СПР: распознавания, логического вывода, информационно-справочных, управления, диагностики (технической и медицинской), "живого" моделирования и пр.

Нахождение рейтинга некоторого объекта в экономике, бизнесе, образовании, науке, а также в аспекте личностной оценки — весьма распространенная и важная задача оценки качества, позволяющая обоснованно принимать оптимальные решения.

Важный класс таких задач образуют задачи выбора оптимальной стратегии управления на основе оценки риска, сопутствующего принятию того или иного решения. Попытка "втянуть" значения многообразных факторов (если они не имеют лишь качественных значений) в одну формулу нахождения риска практически не достигает успеха. Действенными являются лишь экспертные или статистические оценки как значений факторов, так и значения риска.

Например, система управления надежностью инфраструктуры и подвижного состава на железнодорожном транспорте при ограниченном финансировании требует принятия оптимального решения о распределении средств на ремонт, модернизацию

и развитие подструктур и объектов [9]. Для весьма ограниченного числа факторов предложены математические зависимости и оценки, на основе которых ЛПР (лицо, принимающее решение) может обосновать рекомендацию. Однако на большом числе учитываемых факторов на основе нахождения оценок риска дальнейшей эксплуатации транспортных подсистем решение должно приниматься с помощью логических нейронных сетей (составленных для этих подсистем), где риск подобен рейтингу.

Структура нейросети и способы обучения

Для определенности, доходчивости и демонстрации общности построений рассмотрим систему банковского мониторинга, достаточно отображенную в литературе. В работе [10] при построении программного приложения для решения этой задачи предлагается использовать *Карты Кохонена*. Используется "классическая" нейронная сеть, в которой удалось использовать только семь показателей: собственный капитал, сальдированные активы, ликвидные активы, обязательства до востребования, вклады населения, коэффициент ликвидности, бюджетные средства. Фантазируя, можно развить этот ряд показателей. Ими могут быть: объем инвестиций в развивающуюся экономику, объем прибыли, прошлый рейтинг и значение его дрейфа, отчисления в фонд поддержки науки и образования, налоговые отчисления, отчисления в пенсионный фонд, отчисления в благотворительный и культурный фонды, участие в программах ЮНЕСКО и т. д.

Возможны два способа мониторинга банков.

Первый способ предполагает наличие нескольких банков-эталонов, для которых известны их показатели и значения рейтинга, определенные экспертами. На экране создаются концентрические области одинакового рейтинга, занимаемые точками — образами банков-эталонов. По показателям (возможно, нечетким) анализируемого банка методом ассоциативного поиска по близости к известным точкам находится область рейтинга, которой он принадлежит. В этом случае логическая нейронная сеть отражает связь банков-эталонов с их образами — точками, расположенными в рейтинговых областях. Важно, что эти точки в рейтинговой области выбираются произвольно.

Второй способ применяется, если известны значимые области изменения показателей, по принадлежности к которым в совокупности банку может быть присвоен тот или иной рейтинг. В этом случае логические функции, лежащие в основе мониторинга, связывают конъюнкции логических значений (возможно объединение некоторых из них операцией дизъюнкции, если это не приводит к противоречию) принадлежности показателей определенным диапазонам со значением рейтинга. Показатели анализируемого банка могут быть предполагаемыми, неточными.

Напоминаем, что в обоих случаях нейронная сеть является однослойной, содержащей входной слой рецепторов и выходной слой, указывающий на значение координат или рейтинга. Выходной слой должен указывать не только на рейтинг, но и выдавать экспертные рекомендации, а также другие решения и выводы. Таким образом, нейроны своим возбуждением указывают на некоторые тексты, выдаваемые пользователю.

Целесообразен простейший вид обучения (т. е. построения базы знаний, какой является сеть), определяемый концепцией создания нейросети "под задачу". Он заключается в непосредственном введении связей оператором-исследователем "вручную" — от рецепторов к нейронам выходного слоя в соответствии с причинно-следственными связями. Связи отображаются в матрице следования. Таким образом, сеть создается уже обученной, готовой к развитию. Ее функция активации достаточно проста. Ее главная составляющая основана на суммировании величин возбуждения на входе i -го нейрона, умноженных на вес связи. Для реализации первого способа используем функцию активации:

$$V := \frac{\sum_j \omega_j V_j}{\sum_j \omega_j};$$
$$V_i = \begin{cases} V, & \text{если } V \geq h, \\ 0 & \text{в противном случае.} \end{cases}$$

Здесь V — искомое значение возбуждения i -го нейрона; V_j — значение возбуждения j -го нейрона (в данном случае — рецептора), связанного с i -м; ω_j — вес этой связи. (Значение j "пробегаёт" по всем нейронам-рецепторам, связанным с данным так, что значение V , в соответствии с известной формулой нахождения математического ожидания, определяет среднее значение сигнала, поступившего на i -й нейрон).

Применение весов позволяет пользователю в разной степени оперативно учитывать и изменять влияние различных показателей в процессе эксплуатации.

Порог h выбирается экспериментально для подавления слабых возбуждений и упрощения дальнейшей обработки результатов при включении логической нейронной сети в длинные логические цепочки.

Применение матрицы следования позволяет оперативно вносить изменения и уточнения оператором-экспертом-пользователем, развивать сеть, вводя новые показатели (факторы), изменять веса на основе опыта, изменять тексты принимаемых решений.

Структура экрана рецепторов

Могут использоваться два подхода к организации интерфейса: для пользователя, которому важно научиться нажимать нужные кнопки, и для пользователя-оператора, творчески ведущего и развивающего СПР.

При первом подходе экран рецепторов может быть, например, таким, как показан на рис. 1.

Основную часть его составляет "окно прокрутки", через которое можно просматривать и задавать состояние рецепторного слоя, несомненно, не способного поместиться на статическом экране.

В "окне прокрутки" указаны показатели и их оценочные значения (в диапазоне [0, 1]), задаваемые соответствующим рецепторам. Эти значения — вероятностные, по достоверности, интуитивные, экспертные. Они могут предполагать охват нескольких рецепторов. Например, оценка того, что собственный капитал составляет не то 24, не то 34, не то 42 тыс. у. е., но скорее, все-таки, 24, может привести к приблизительной оценке задаваемых величин возбуждения, равных 0,6, 0,2 и 0,2 (в сумме равных единице) соответственно рецепторов, "ответчающих" за диапазоны (20—25], (30—35], (40—45].

На экране отображены задаваемые без прокрутки показатели, такие как рейтинг в результате прошлых измерений, выборочные "прошлые" показатели, а также показатели политической, социальной и экономической конъюнктуры. (Их обилие и развитие могут все-таки потребовать прокрутки.)

Должно быть отображено управление прокруткой, а также меню основных действий. Эти действия следующие: переход на экран выходного слоя, статистическая обработка результатов (предполагает переход к экрану выходного слоя), введение новой связи, введение нового рецептора, изменение веса связи, введение нового нейрона выходного слоя.

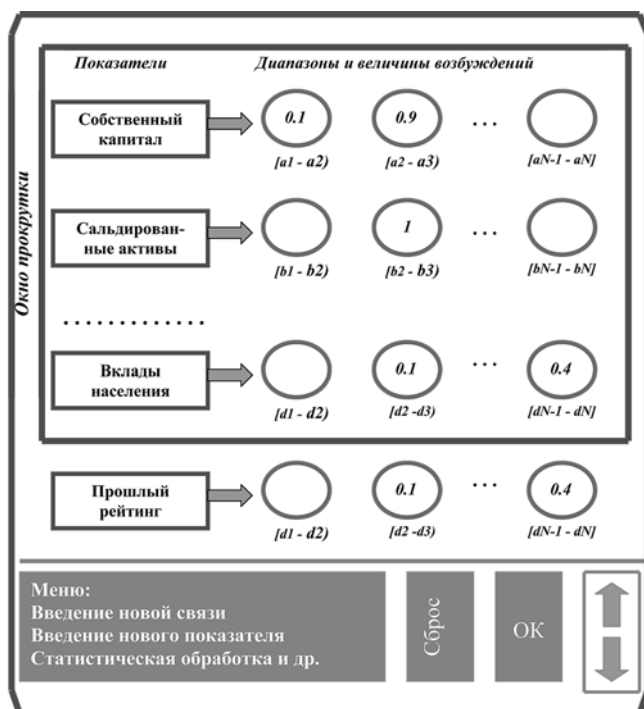


Рис. 1. Экран рецепторов

Структура экрана выходного слоя, отображающего рейтинг банков-эталонов

Экран выходного слоя (рис. 2, см. четвертую сторону обложки) может отображать систему вложенных прямоугольников или других плоских фигур, отражающих распространение рейтинга по убыванию. В центре экрана яркими точками отражены самые преуспевающие банки или предполагаемые идеальные образы таких банков. Каждый элемент экрана жестко закреплен за нейроном выходного слоя. В результате мониторинга по принципу ассоциативного мышления на экране должна появиться точка, соответствующая анализируемому банку. По ее попаданию в область рейтинга известных банков или по совпадению этой точки с точкой известного эталона следует судить о значении рейтинга банка.

Обычно в системах компьютерной графики точка начала координат расположена в левом нижнем углу экрана монитора. Тогда для ее переноса в центр с учетом "сплющивания" по оси ординат необходимо выполнить преобразование:

$$x = x^* - x_0,$$

$$y = k(y^* - y_0).$$

Здесь x, y — декартовы координаты точки экрана рейтинговых областей; x^*, y^* — декартовы координаты этой точки в используемой графической системе; x_0, y_0 — координаты центра экрана в этой системе; k — коэффициент "сплющивания" экрана по оси ординат.

Однако экран, связанный с декартовой системой координат, как следует из рис. 2 (см. четвертую сторону обложки), затрудняет интерполяцию, необходимую при ассоциативном выводе. А именно, найденная на основе усреднения координата двух точек, принадлежащих некоторому, предположим, — низкому, рейтингу, может оказаться между этими точками и принадлежать области высокого рейтинга.

Для облегчения интерполяции целесообразно в качестве рабочей использовать сферическую систему координат $Or\varphi$ (рис. 3, см. четвертую сторону обложки), связанную с декартовыми соотношениями:

$$r = \sqrt{x^2 + y^2}, \quad \varphi = \text{arctg} \frac{y}{x}.$$

Центр O совпадает с центром экрана. Тогда области одинакового рейтинга оказываются ограниченными сферическими эллипсами, как показано на рис. 3 (см. четвертую сторону обложки).

Поясним размещение на рис. 3 точек B_1, \dots, B_6 (точка B рассмотрена далее). Пусть в декартовой системе координат известны и произвольно распределены по областям одинакового рейтинга точки, соответствующие банкам: $B_1(6, 6), B_2(8, 5), B_3(3, 7), B_4(7, 3), B_5(12, 6), B_6(4, 1)$. Тогда в сферической системе координат на рис. 3 эти точки определяются как: $B_1(3, 190^\circ), B_2(3, 300^\circ), B_3(7, 45^\circ), B_4(8, 250^\circ), B_5(12, 210^\circ), B_6(11, 80^\circ)$.

Пример определения рейтинга на основе банков-эталонов

Все рекомендуемые построения проиллюстрируем простым примером, не претендующим на правильность применения в финансовой деятельности. Пусть Экспертный Совет выделил четыре показателя (фактора) для банковского мониторинга:

- z_1 — собственный капитал;
- z_2 — вклады населения;
- z_3 — объем вложений в культурные программы ЮНЕСКО;
- z_4 — объем прибыли.

Рейтинг банка может быть: R_1 — высокий, R_2 — средний, R_3 — низкий. При этом вполне справедливо учитывать показатель z_3 с весом 0,5.

Тогда каждому банку B соответствует вектор его показателей $B(z_1, z_2, z_3, z_4)$, лежащий в основе определения его рейтинга.

Пусть известен рейтинг ряда крупных международных банков на основе их показателей:

- $B_1(\$45 \text{ млрд}; \$25 \text{ млрд}; \$10 \text{ млрд}; \$5 \text{ млрд}) \rightarrow R_1;$
- $B_2(\$25 \text{ млрд}; \$12 \text{ млрд}; \$5 \text{ млрд}; \$2 \text{ млрд}) \rightarrow R_1;$
- $B_3(\$20 \text{ млрд}; \$15 \text{ млрд}; \$2 \text{ млрд}; \$3 \text{ млрд}) \rightarrow R_2;$
- $B_4(\$10 \text{ млрд}; \$12 \text{ млрд}; \$6 \text{ млрд}; \$1 \text{ млрд}) \rightarrow R_2;$
- $B_5(\$20 \text{ млрд}; \$1 \text{ млрд}; \$0 \text{ млрд}; \$0 \text{ млрд}) \rightarrow R_3;$
- $B_6(\$1 \text{ млрд}; \$0,5 \text{ млрд}; \$0 \text{ млрд}; \$0,1 \text{ млрд}) \rightarrow R_3.$

Эти соотношения определяют исходное содержание базы знаний (БЗ), которое может дополняться в ходе эксплуатации.

На рис. 4 представлена логическая нейронная сеть, соответствующая данной БД. Рецепторы 1—21 в соответствии со значением достоверности приведенных при них высказываний указывают на координаты точек на экране, соответствующих банкам-эталонам. Единичные веса не обозначены.

По предполагаемому, возможно, неточным показателям методом ассоциативных вычислений находится точка, соответствующая анализируемому банку. По области ее попадания делается вывод о рейтинге этого банка.

Пример 1. Найдем рейтинг банка B по предполагаемым показателям ($\$28 \text{ млрд}; \$3 \text{ млрд}; \$8 \text{ млрд}; \4 млрд)

Приведем общий прием возбуждения рецепторов. А именно, для возбуждения рецепторов, закрепленных за каждым показателем, делим "единицу" обратно пропорционально расстоянию между двумя ближайшими значениями этого показателя, отображенными в сети. Тогда возбуждение V_4 рецептора 4 приблизительно полагаем равным 0,9, а возбуждение V_5 рецептора 5 — 0,1. Аналогично, приблизительно полагаем $V_7 = 0,8$, $V_8 = 0,2$, $V_{14} = V_{15} = V_{20} = V_{21} = 0,5$. Возбуждение остальных рецепторов равно нулю.

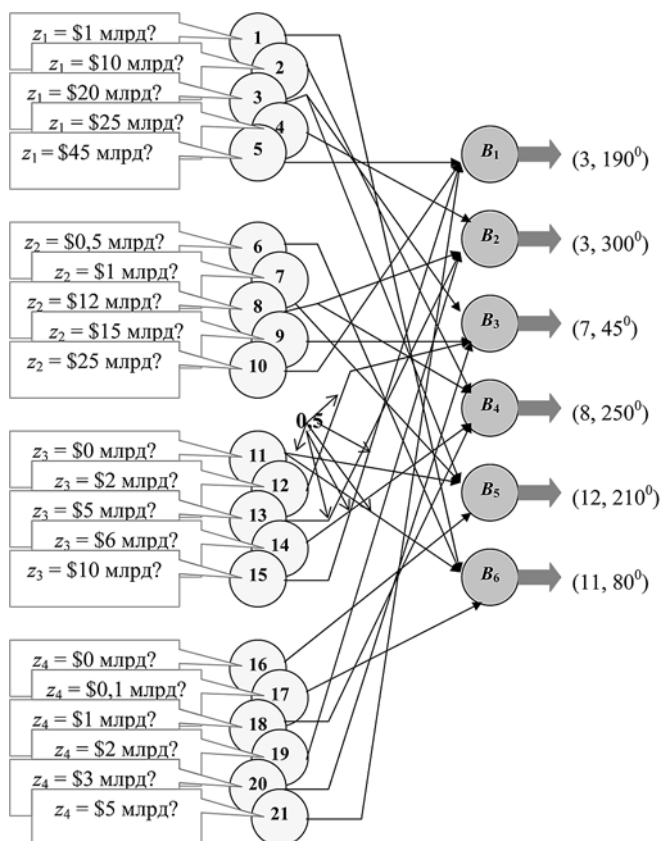


Рис. 4. Логическая нейронная сеть на базе рейтинга известных банков-эталонов

По функции активации при $h = 0,2$ (значение порога подбирается так, чтобы исключить из рассмотрения малые возбуждения) находим возбуждение нейронов, соответствующих точкам банков-эталонов на экране: $V_{B1} = 0,24$, $V_{B2} = 0,46$, $V_{B3} = V_{B4} = 0$, $V_{B5} = 0,23$, $V_{B6} = 0$. Используя полученные значения возбуждения нейронов в качестве весов, по известной формуле нахождения среднего найдем значение координат искомой точки B на экране:

$$r = \frac{3 \cdot 0,24 + 3 \cdot 0,46 + 12 \cdot 0,23}{0,97} \approx 5,0;$$

$$\varphi = \frac{190^\circ \cdot 0,24 + 300^\circ \cdot 0,46 + 210^\circ \cdot 0,23}{0,97} \approx 240^\circ.$$

На рис. 3 (см. четвертую сторону обложки) показано примерное положение найденной точки B на экране. Она принадлежит области среднего рейтинга R_2 .

Построение обученной логической нейронной сети на основе интервалов значений показателей

Обучение рейтинговой системы с помощью известных рейтингов банков, принимаемых за эталон, приводит, как показано выше, к достаточно громоздким логическим нейронным сетям. Ведь они должны развиваться с обретением новых знаний.

Следует учесть, однако, высокую степень наглядности и "дружественный" интерфейс. Несомненно, более точно и оперативно используется накопленный опыт.

Можно создавать значительно более простые логические нейронные сети, если известны диапазоны принадлежности значений показателей, в совокупности определяющие то или иное значение рейтинга.

Например, пусть на основе изучения рейтинга приведенных выше банков и значительного числа других оказалось целесообразным выделить по два значимых диапазона возможной принадлежности каждого показателя банка. Пусть z_1 определяется отрезками (возможно, полуоткрытыми) значений $[0, 25]$ и $[25, 50]$, z_2 — отрезками $[0, 10]$ и $[10, 25]$, z_3 — отрезками $[0, 5]$ и $[5, 10]$, z_4 — отрезками $[0, 2]$ и $[2, 5]$.

На основе информации экспертов составляется логическое описание системы принятия решений. В соответствии с результатами исследований, отображенных в приведенной литературе, оно должно состоять из исчерпывающего, непротиворечивого набора логических функций — импликаций. Каждая логическая функция в левой части представляет собой конъюнкцию, в общем случае, дизъюнкций высказываний о принадлежности данных. То есть в левой части отображается ситуация. В правой части, после знака следования, указывается значение рейтинга. Таким образом, правые части отображают прием "размножения решений", способствующий непротиворечивости и возможности модификации.

При этом для сокращения объема сети в левых частях логических функций можно использовать объединение данных одного фактора (показателя) в дизъюнкции так, чтобы при раскрытии скобок не сформировались ситуации, для которых предусмотрено решение, отличное от указанного в правой части. Это означает, что при неправильном формировании логического описания одни ситуации могут поглотить другие, требующие отличающегося решения.

В то же время попытки сократить объем логической нейронной сети за счет объединения ситуаций, требующих одного и того же решения, с помощью операции дизъюнкции вполне справедливы.

Пусть логическое описание, удовлетворяющее указанным требованиям полноты и непротиворечивости, имеет вид:

$$(z_1 \in [25, 50]) \wedge (z_2 \in [10, 25]) \wedge ((z_3 \in [0, 5]) \vee \vee (z_3 \in [5, 10])) \wedge (z_4 \in [2, 5]) \rightarrow Q_1 = \langle \text{Рейтинг банка высокий} \rangle;$$

$$(z_1 \in [0, 25]) \wedge (z_2 \in [10, 25]) \wedge ((z_3 \in [5, 10]) \wedge \wedge (z_4 \in [2, 5])) \rightarrow Q_2 = \langle \text{Рейтинг банка средний} \rangle;$$

$$(z_1 \in [25, 50]) \wedge (z_2 \in [10, 25]) \wedge ((z_3 \in [0, 5]) \vee \vee (z_3 \in [5, 10])) \wedge (z_4 \in [0, 2]) \rightarrow Q_3 = \langle \text{Рейтинг банка средний} \rangle;$$

$$(z_1 \in [25, 50]) \wedge (z_2 \in [10, 25]) \wedge (z_3 \in [0, 5]) \wedge \wedge (z_4 \in [2, 5]) \rightarrow Q_4 = \langle \text{Рейтинг банка средний} \rangle;$$

$$(z_1 \in [0, 25]) \wedge (z_2 \in [0, 10]) \wedge ((z_3 \in [0, 5]) \vee \vee (z_3 \in [5, 10])) \wedge (z_4 \in [0, 2]) \rightarrow Q_5 = \langle \text{Рейтинг банка низкий} \rangle.$$

Обученная логическая нейронная сеть, напрямую связывающая диапазоны показателей с рейтингом банка и учитывающая веса связей (0,5 — вес всех связей, исходящих из вершин 5 и 6), показана на рис. 5. Рецепторы 1—8 возбуждаются в соответствии с достоверностью указанных при них высказываний. Легко видеть, что рассмотренные ранее банки-эталон точно отражены в этой сети.

Однако использованная ранее функция активации здесь не подходит, так как на некоторых нейронах объединяется не единственная конъюнкция событий, а дизъюнкция таких конъюнкций. Целесообразно выбрать следующую функцию активации:

$$V := \sum_j \omega_j V_j;$$

$$V_i = \begin{cases} V, & \text{если } V \geq h, \\ 0 & \text{в противном случае.} \end{cases}$$

Здесь, как и прежде, V_i — искомое значение возбуждения i -го нейрона; V_j — значение возбуждения j -го нейрона (рецептора), связанного с i -м; ω_j — вес этой связи.

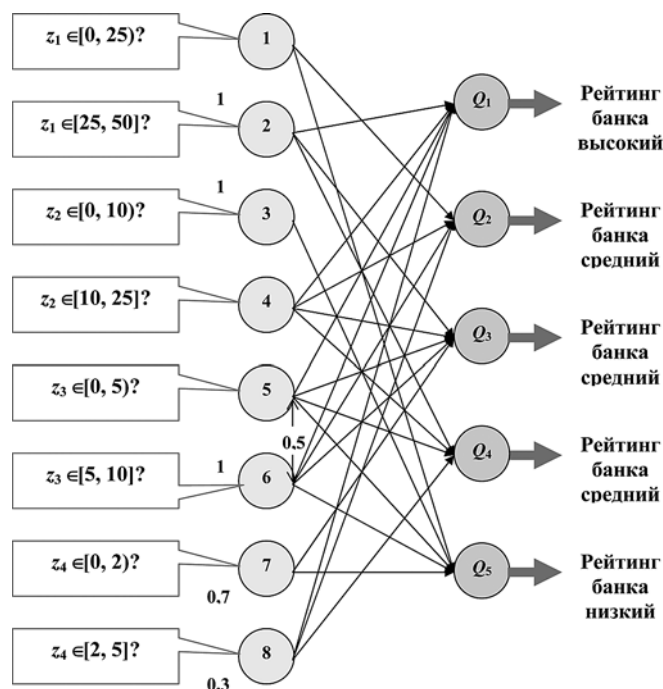


Рис. 5. Логическая нейронная сеть, связывающая диапазоны значений показателей с рейтингом банка

Если предполагается данную сеть использовать в длинной логической цепочке, где значения возбуждения нейронов тотчас поступают на рецепторы другой логической нейронной сети, то необходимо выполнить нормирование формируемых возбуждений. Тогда значение V следует разделить на максимально возможное значение возбуждения нейрона. В данном примере это значение равно 3,5.

Пример 2. Определим рейтинг банка B , рассмотренного в примере 1, с учетом того, что показатель z_4 (объем прибыли) определен неточно и составляет не то \$1,5 млрд, не то \$4 млрд. Размышляя трезво, аналитик предположил, что первая оценка верна на 70 %, а вторая, соответственно, на 30 %. Зададим возбуждение рецепторов: $V_2 = 1$, $V_3 = 1$, $V_6 = 1$, $V_7 = 0,7$, $V_8 = 0,3$. (Значения отмечены на рис. 5.) С помощью функции активации, положив $h = 1$, находим $V_{Q1} = 1,8$, $V_{Q2} = 0$, $V_{Q3} = 2,2$, $V_{Q4} = 1,3$, $V_{Q5} = 2$. Так как максимального возбуждения достигает нейрон Q_3 , банк B обладает средним рейтингом.

Возникает лишь вопрос об отображении полученной сети на экране монитора.

Отображение и управление рейтинговой системой с помощью матриц следования

Существует универсальный, практически реализуемый способ представления и обработки логических нейронных сетей с помощью матриц следования, отображенных на экране монитора. Он использует лишь простейшие средства компьютерной обработки таблиц.

Матрица следования, отображающая представленную на рис. 5 логическую нейронную сеть, показана в табл. 1. Она является основным средством управления рейтинговой системой, обеспечивая ее развитие и модификацию. Это средство дополняет наглядный, информативный экран, являющийся эффективным средством визуализации.

Однако матрицу следования логической нейронной сети, связывающей диапазоны значений показателей с рейтингом банка (табл. 2), не следует считать менее информативным и оперативным средством управления, модификации и развития рейтинговой системы, несмотря на меньший эффект художественного восприятия.

Таблица 1

Матрица следования логической нейронной сети на базе рейтинга известных банков

Показатели банков-эталонов	$z_1 = \$1$ млрд	$z_1 = \$10$ млрд	$z_1 = \$20$ млрд	$z_1 = \$25$ млрд	$z_1 = \$45$ млрд	$z_2 = \$0,5$ млрд	$z_2 = \$1$ млрд	$z_2 = \$12$ млрд	$z_2 = \$15$ млрд	$z_2 = \$25$ млрд	$z_3 = \$0$ млрд	$z_3 = \$2$ млрд	$z_3 = \$5$ млрд	$z_3 = \$6$ млрд	$z_3 = \$10$ млрд	$z_4 = \$0$ млрд	$z_4 = \$0,1$ млрд	$z_4 = \$1$ млрд	$z_4 = \$2$ млрд	$z_4 = \$3$ млрд	$z_4 = \$5$ млрд	Возбуждение нейронов
	Задаваемые значения			0,9	0,1		0,8	0,2						0,5	0,5						0,5	
$B_1(3, 190^\circ)$					1					1					0,5						1	0,24
$B_2(3, 300^\circ)$				1				1					0,5						1			0,46
$B_3(7, 45^\circ)$			1						1			0,5								1		0
$B_4(8, 250^\circ)$		1						1					0,5				1					0
$B_5(12, 210^\circ)$			1				1				0,5					1						0,23
$B_6(11, 80^\circ)$	1					1					0,5						1					0

Таблица 2

Матрица следования логической нейронной сети, связывающая диапазоны значений показателей с рейтингом банка

Интервалы показателей банков	$z_1 \in [0, 25]$	$z_1 \in [25, 50]$	$z_2 \in [0, 10]$	$z_2 \in [10, 25]$	$z_3 \in [0, 5]$	$z_3 \in [5, 10]$	$z_4 \in [0, 2]$	$z_4 \in [2, 5]?$	Возбуждение нейронов
Задаваемые оценки принадлежности		1	1			1	0,7	0,3	
$R_1 = \langle \text{Рейтинг высокий} \rangle$		1		1	0,5	0,5		1	1,8
$R_2 = \langle \text{Рейтинг средний} \rangle$	1			1		0,5		1	0
$R_3 = \langle \text{Рейтинг средний} \rangle$		1		1	0,5	0,6	1		2,2
$R_4 = \langle \text{Рейтинг средний} \rangle$		1		1	0,5			1	1,3
$R_5 = \langle \text{Рейтинг низкий} \rangle$	1		1		0,5	0,5	1		2

Заклучение

Унифицированный программный продукт на основе рассмотренных выше логических нейронных сетей должен быть инвариантным относительно числа рецепторов и числа нейронов выходного слоя. Изменение этих параметров пользователем-оператором с помощью матриц следования, а также изменение конфигурации связей могут проводиться оперативно. Такой же оперативной, доступной и изменяемой по желанию пользователя, является информация о смысловом содержании запросов и решений. Примененные функции активации соответствуют универсальности логической нейронной сети.

Применение логических нейронных сетей в рейтинговых системах весьма результативно в различных системах приоритетного планирования, управления качеством, оценки результатов трудовой деятельности, в социальных системах, в образовании — во всех системах и приложениях, где точная математическая интерпретация процессов невозможна или весьма трудоемка.

Список литературы

1. Барский А. Б. Нейронные сети и искусственный интеллект // Информационные технологии. Приложение. 2003. № 1. 32 с.

2. Барский А. Б. Логические нейронные сети: методика построения и некоторые применения // Информационные технологии. Приложение. 2006. № 8. 32 с.

3. Барский А. Б. Математическая логика событий и логические нейронные сети // Информационные технологии. Приложение. 2007. № 7. 32 с.

4. Барский А. Б., Дмитриев А. А., Барская О. А. Медицинские информационно-справочные системы на логических нейронных сетях // Информационные технологии. Приложение. 2010. № 1. 32 с.

5. Барский А. Б. Нейросетевые технологии на транспорте // Мир Транспорта. 2011. № 2. С. 4—11, № 3. С. 14—19.

6. Барский А. Б. Нейронные сети: Распознавание, управление, принятие решений. М.: Финансы и статистика. 2004. 175 с.

7. Барский А. Б. Логические нейронные сети. М.: ИНТУИТ; БИНОМ. Лаборатория знаний. 2007. 352 с.

8. Барский А. Б. Нейронные сети логического вывода. Курс лекций. Saarbrücken: LAP LAMBERT Academic Publishing. 2011. 400 с.

9. Новиков П. С. Элементы математической логики. М.: Гос. изд-во физико-мат. литературы, 1959. 400 с.

10. Кобринский Н. С., Трахтенброт Б. А. Введение в теорию конечных автоматов. М.: Гос. изд-во физико-мат. литературы, 1962. 404 с.

11. Кемени Дж., Снелл Дж., Томпсон Дж. Введение в конечную математику: Пер. с англ. М.: Изд-во иностр. лит., 1963. 486 с.

12. Замышляев А. М. Прикладные информационные системы управления надежностью, безопасностью, рисками и ресурсами на железнодорожном транспорте // Надежность. 2013. 143 с.

13. Deboeck G., Kohonen T. Visual Exploration in Finance with Self-Organizing Maps. Springer-Velag, 1998.

A. B. Barsky, Professor, e-mail: arkbarsk@mail.ru,
Moscow State University of Railway Engineering (MIIT)

The Rating Systems on Base of the Logical Neural Networks

The possibility of the logic neural networks applying in the rating system on base of the key indicators founded with the uncertainty is investigated. The technology is applicable for bank monitoring, monitoring in education, in political and economic systems, to determine the risk country, insurance, military-strategic, in the management of reliability of complex systems, etc. "Non-formula" associative computing are based on the knowledge base what the experimental, expert or exploitation data accumulated. The one-level logical neural network is using, producing "voting" with a very simple threshold activation function. A simplified model of bank monitoring demonstrates two approaches: ranking of the bank controlled with help of etalon banks clusters, generators rated area and finding rating based on an assessment of individual indicators toiletries certain ranges of variation, together with some precision in determining the rating of the bank. The recommendations on the organization of the user's screen are giving. Studies are orienteering to developing in unified software applications for a wide range of "rating" problems.

Keywords: logic neural network, the activation function, the matrix sequence, a rating system monitoring

References

1. Barskij A. B. Nejrionnye seti i iskusstvennyj intellekt. *Informacionnye tehnologii. Prilozhenie*. 2003. N. 1. 32 p.

2. Barskij A. B. Logicheskie nejrionnye seti: metodika postroenija i nekotorye primenenija. *Informacionnye tehnologii. Prilozhenie*. 2006. N. 8. 32 p.

3. Barskij A. B. Matematicheskaja logika sobytij i logicheskie nejrionnye seti. *Informacionnye tehnologii. Prilozhenie*. 2007. N. 7. 32 p.

4. Barskij A. B., Dmitriev A. A., Barskaya O. A. Medicinskie informacionno-spravochnye sistemy na logicheskix nejrionnyh setyah. *Informacionnye tehnologii. Prilozhenie*. 2010. N. 1. 32 p.

5. Barskij A. B. Nejrosetevye tehnologii na transporte. *Mir Transporta*. 2011. N. 2. P. 4—11, N. 3. P. 14—19.

6. Barskij A. B. Nejrionnye seti: raspoznavanie, upravlenie, prinyatie reshenij. M.: Finansy i statistika. 2004. 175 p.

7. Barskij A. B. Logicheskie nejrionnye seti. M.: INTUIT, BINOM. Laboratoriya znaniy. 2007. 352 p.

8. Barskij A. B. Nejrionnye seti logicheskogo vyvoda. Kurs lekcij. Saarbrücken: LAP LAMBERT Academic Publishing. 2011. 400 p.

9. Novikov P. S. Elementy matematicheskoy logiki. M.: Gos. izd-vo fiziko-mat. literatury, 1959. 400 p.

10. Kobrinskij N. S., Trahtenbrot B. A. Vvedenie v teoriyu konechnyh avtomatov. M.: Gos. izd-vo fiziko-mat. literatury, 1962. 404 p.

11. Kemeni Dzh., Snell Dzh., Tompson Dzh. Vvedenie v konechnuyu matematiku: Per. s angl. M.: Izd-vo inostr. lit. 1963. 486 p.

12. Zamyshlyayev A. M. Prikladnye informacionnye sistemy upravleniya nadyozhnost'yu, bezopasnost'yu, riskami i resursami na zheleznodorozhnom transporte. M.: Nadyozhnost'. 2013. 143 p.

13. Deboeck G., Kohonen T. Visual Exploration in Finance with Self-Organizing Maps. Springer-Velag, 1998.

В. И. Аникин, д-р техн. наук, проф., e-mail: anikin_vi@mail.ru,
Поволжский государственный университет сервиса, г. Тольятти,
А. А. Карманова, инженер-программист, e-mail: turaeva.alexandra@mail.com,
ООО "НетКрэкер", г. Тольятти

Обучение искусственной нейронной сети Кохонена клеточным автоматом

Изучены закономерности обучения искусственной нейронной сети Кохонена с помощью клеточного автомата. Предложен оригинальный метод инициализации и начального разворачивания нейронов сети, предотвращающий скручивание самоорганизующейся карты Кохонена в итерациях обучения. Продемонстрирована возможность решения известной проблемы "мертвых" нейронов путем использования обучающего клеточного автомата на многосвязной карте Кохонена.

Ключевые слова: нейронная сеть, Кохонен, клеточный автомат, алгоритм обучения, Excel

Введение

Обучение искусственной нейронной сети (ИНС) Кохонена относится к типу самообучающегося соревнования. Это явление, известное как правило "Победитель забирает все" (Winner Takes All — WTA), имеет нейрофизиологическую аналогию [1]. Подстройка весов связей в алгоритме WTA осуществляется не для всех нейронов слоя Кохонена, а только для нейрона-победителя, значение выхода которого в слое нейронов оказывается максимальным.

Главная проблема алгоритма WTA — наличие "мертвых" нейронов, которые ни разу не побеждают в конкурентной борьбе. Для разрешения этой проблемы применяются модифицированные методы и алгоритмы самообучения, дающие лучшие результаты, чем алгоритм WTA.

Алгоритм "Победитель справедливо получает все" (Conscience Winner Takes All — CWTA) [2] является улучшенной модификацией алгоритма WTA. Он позволяет учитывать число побед и поощрять нейроны с наименьшей активностью для выравнивания их шансов на победу. Этот алгоритм считается одним из лучших и быстрых в классе алгоритмов самоорганизации.

В алгоритме "Победитель получает больше" (Winner Takes Most — WTM) [2] кроме нейрона-победителя обучаются также нейроны из его ближайшего окружения. При этом чем дальше какой-либо нейрон находится от победителя, тем меньше изменяются его веса.

В алгоритме "Изменяющаяся во времени самоорганизующаяся карта" (Time Adaptive Self-Organizing Map — TASOM) [3] каждый нейрон имеет свою собственную скорость обучения и функцию соседства. Для каждого нового входного вектора размер окрестности, скорость обучения победившего нейрона и скорость обучения соседних нейронов обновляются в целях адаптации к изменяющейся среде.

Общим недостатком рассмотренных алгоритмов обучения нейронной сети Кохонена является наличие в них большого числа эвристических параметров и искусственность предлагаемых решений проблемы "мертвых" нейронов.

В работе [4] авторами данной статьи предложен и теоретически обоснован оригинальный алгоритм обучения ИНС Кохонена клеточным автоматом с радиусом окружения $r = 1$, обеспечивающий простую инициализацию и эффективное обучение нейронной сети на максимальной скорости. Автор статьи [5] отмечает близкое сходство явлений самоорганизации, наблюдаемых в нейронной сети Кохонена [1] и клеточных нейронных сетях (Cellular Neural Networks — CNN). В связи с этим уместно заметить, что конечным следствием предложенного нами алгоритма обучения является обобщение классической нейронной сети Кохонена в клеточную нейронную сеть соревновательного типа.

Обоснование и постановка задачи исследования

Целью работы является систематическое экспериментальное исследование авторского алгоритма обучения ИНС Кохонена клеточным автоматом [4], дискретным в пространстве и во времени, но непрерывным по состояниям. Такого рода непрерывные клеточные автоматы, введенные в употребление К. Канеко в 1984 г. [6], получили название решеток связанных отображений (*coupled map lattices*).

Исследуемый алгоритм обучения ИНС Кохонена клеточным автоматом является одним из вариантов пакетного алгоритма обучения сети [1]. Его главными особенностями являются простота реализации, гибкость и отличная от классической нестандартная интерпретация физических механизмов, обеспечивающих эффективное обучение нейронной сети.

Ячейкой клеточного автомата в нашем алгоритме является один нейрон слоя Кохонена, в окружении

фон Неймана топологически связанный с четырьмя (на границе — с двумя или тремя), а в окружении Мура — с восемью (на границе — с тремя или пятью) соседними нейронами двумерной самоорганизующейся карты Кохонена. Соответственно, клеточный автомат — это множество взаимодействующих со своими соседями нейронов слоя Кохонена, помещенных в непрерывное пространство обучающих образцов.

Состояние ячейки клеточного автомата задается многомерным вектором, компоненты которого суть координаты соответствующего нейрона в пространстве обучающих образцов. В соответствии с принципом "ближайшего соседа" нейроны сети являются центрами многомерных окрестностей Вороного, на которые, подобно мозаике, делится все пространство учебных образцов в итерациях обучения.

Обучение сети выполняется в пакетном режиме, т. е. клеточный автомат переходит в новое состояние один раз в конце каждой эпохи после подачи на входы сети всех учебных образцов, которые могут выбираться в произвольном порядке, например последовательно.

Согласно предложенному алгоритму обучения любой k -й нейрон карты Кохонена в итерациях эпохи может изменить свой многомерный вектор состояния в двух случаях: 1) нейрон стал победителем; 2) нейрон является соседом другого победившего нейрона. Тогда среднее суммарное приращение i -й компоненты вектора состояния k -го нейрона с окружением фон Неймана¹ за одну эпоху будет равно

$$\Delta w_{ki}(t) = \eta \left[\sum_{m=j=1}^{n_m} (x_{mi}^{(j)}(t) - w_{ki}(t)) \right] / \left[\sum_{m=k}^{k+4} n_m(t) \right], \quad (1)$$

где η — скорость обучения; k — номер нейрона-победителя; $m = \{k-1, k, k+1, k-L, k+L\}$ — номера нейронов из окружения k -го нейрона; L — ширина прямоугольной самоорганизующейся карты Кохонена; n_m — число учебных образцов, попавших в окрестность Вороного m -го нейрона; $x_{mi}^{(j)}$ — i -я координата j -го учебного образца из окрестности Вороного m -го нейрона; w_{ki} — i -я координата k -го нейрона.

Обозначив $n = \sum_{m=k}^{k+4} n_m$, выражение (1) можно переписать в виде

$$\Delta w_{ki} = \eta \frac{1}{n} \left[\sum_{m=k}^{k+4} n_m (\bar{x}_{mi} - w_{ki}) \right], \quad (2)$$

¹Аналогичные формулы можно записать и для других структур окружения КА.

где \bar{x}_{mi} — i -я координата центра тяжести окрестности Вороного m -го нейрона.

Отсюда видно, что в каждой эпохе обучения k -й нейрон смещается на относительную величину η в направлении центра тяжести области пространства учебных образцов, образованной объединением окрестностей Вороного k -го нейрона-победителя и его ближайших соседей. В частности, при $\eta = 1$ нейрон смещается точно в центр тяжести этой объединенной области.

Введение понятия клеточного автомата позволяет при интерпретации закономерностей обучения ИНС Кохонена пользоваться терминологией и выводами теории клеточных автоматов, такими как [7]:

- *структура окружения* (в частности, окружения Мура и фон Неймана); *глобальное состояние клеточного автомата*, *функция переходов*, определяющая итерационный процесс эволюции клеточного автомата; *аттракторы и бассейны притяжения*, достигаемые клеточным автоматом из заданной начальной конфигурации [8] и др.;
- *классы типичных конфигураций клеточного автомата* и, в частности, наблюдаемые при обучении ИНС Кохонена *предельная неподвижная (eventually fixed)* и *предельная периодическая (eventually periodic)* конфигурации [7];
- *глобальная самоорганизация клеточного автомата как результат локальных взаимодействий* между его ячейками. Этот вывод теории клеточных автоматов особенно важен, поскольку он означает, что при обучении ИНС Кохонена клеточным автоматом достаточно учитывать только локальные взаимодействия между соседними нейронами сети, т. е. использовать структуру окружения радиуса 1.

Метод решения задачи исследования

Важной особенностью нашего исследования является то, что ИНС Кохонена и алгоритм ее обучения клеточным автоматом были реализованы в электронной таблице Microsoft Excel чисто табличными средствами, без программирования на языке VBA и использования пользовательских функций, что само по себе является нетривиальной задачей [9]. Важно заметить, что пользовательские функции работают в Excel значительно медленнее встроенных функций, поэтому отсутствие этих функций в табличной модели ИНС Кохонена позволило нам достичь максимального быстродействия при обучении сети.

Пользовательский интерфейс табличной модели нейронной сети Кохонена, обучаемой клеточным автоматом, позволяет легко изменять настраиваемые параметры и визуально наблюдать за ходом обучения сети (рис. 1).

Примечательной особенностью алгоритма обучения нейронной сети Кохонена клеточным авто-

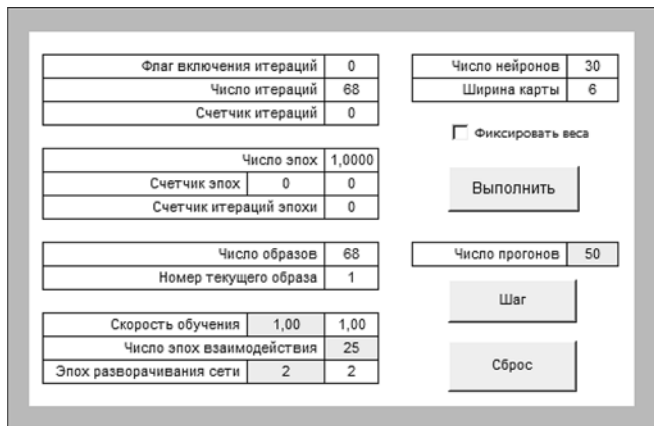


Рис. 1. Пользовательский интерфейс управления и визуализации процесса обучения нейронной сети Кохонена

матом, в отличие от классического алгоритма обучения WTA и его модификаций, является небольшое число настраиваемых параметров, к которым относятся:

- *число прогонов* — число эпох обучения нейронной сети, в каждой эпохе учебные образцы подаются на вход сети последовательно, в порядке возрастания их номеров;
- *скорость обучения* — параметр, регулирующий скорость обучения сети Кохонена клеточным автоматом, в наших экспериментах он всегда был равен 1;
- *число эпох взаимодействия* — число эпох обучения, в которых связи между соседними нейронами сети остаются включенными;
- *число эпох разворачивания сети* — число эпох взаимодействия, в которых нейроны сети из заданной удаленной начальной точки "втягиваются" в гиперкуб нормированного пространства учебных образцов, детально процесс разворачивания нейронов сети описан ниже.

Кнопка *Выполнить* (рис. 1) запускает итерации обучения нейронной сети в течение заданного числа прогонов, кнопка *Шаг* позволяет выполнять обучение сети в шаговом режиме (используется в

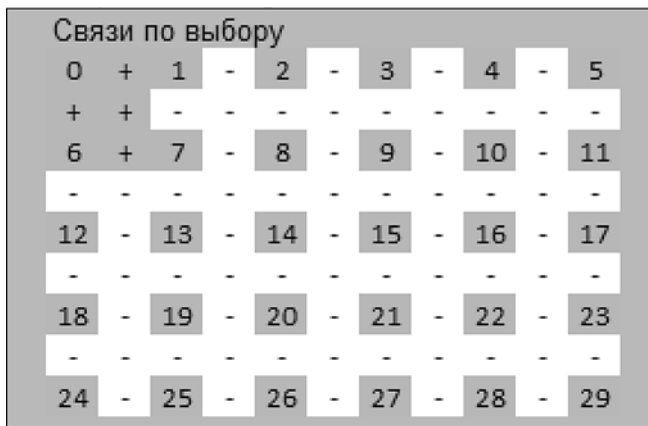


Рис. 2. Интерфейс управления связями между нейронами обучающего клеточного автомата размером 2×2 : "+" — связь включена; "-" — связь выключена

режиме отладки), кнопка *Сброс* приводит ИНС Кохонена в исходное необученное состояние.

Пользовательский интерфейс управления структурой обучающего клеточного автомата показан на рис. 2.

Как видно, табличная модель нейронной сети Кохонена может содержать до 30 связанных нейронов с топологией двумерной решетки размером 5×6 . На рис. 2 активными являются нейроны с номерами 0, 1, 6 и 7, остальные нейроны не задействованы.

На рис. 3 показаны структуры обучающих клеточных автоматов, использованных в обсуждаемых ниже экспериментах с табличной моделью ИНС Кохонена.

Как известно, для успешного обучения ИНС Кохонена принципиальное значение имеет задание начального состояния ее нейронов. Кохонен [1] предложил два способа начальной инициализации нейронов в пакетных алгоритмах обучения: 1) произвольные случайные состояния и 2) любое упорядоченное состояние, например, нейроны располагаются в узлах прямоугольной сетки на плоскости главных компонент пространства учебных образцов.

Способ инициализации сети случайными состояниями нейронов не обеспечивает наискорей-

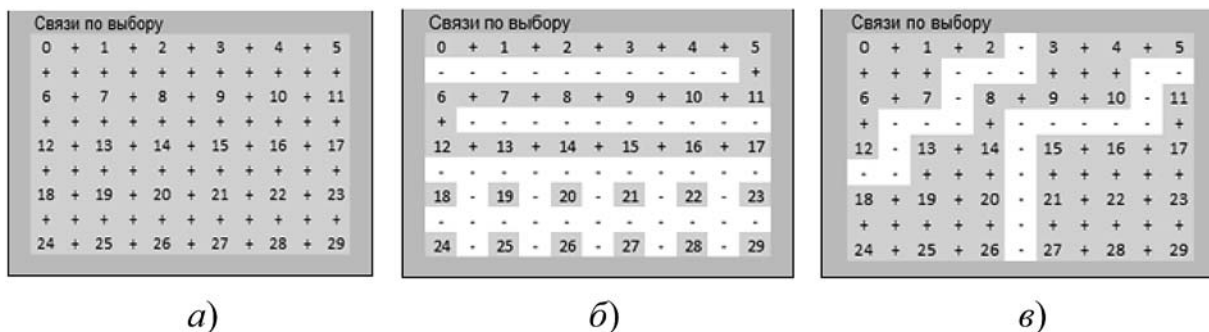


Рис. 3. Структуры обучающих клеточных автоматов:

а — двумерный клеточный автомат с окружением Мура; б — одномерный клеточный автомат; в — трехсвязный клеточный автомат

шего упорядочивания нейронов сети, а при обучении клеточным автоматом часто приводит к скручиванию карты Кохонена. Второй способ инициализации нейронов сети довольно трудоемкий и неоднозначен в смысле выбора местоположения узлов упомянутой сетки.

В нашем алгоритме обучения использован нестандартный способ задания начального состояния нейронов сети, обеспечивающий самостоятельное адаптивное разворачивание нейронов с образованием упорядоченной самоорганизующейся карты Кохонена. Для этого при инициализации сети все нейроны исходно помещаются в произвольную точку на гиперсфере достаточно большого радиуса R , центр которой совпадает с центром пространства учебных образцов. Заметим, что пространство учебных образцов в нашем алгоритме нормировано и образует гиперкуб размерностью d с длиной ребра, равной 2.

Радиус R гиперсферы выбирается таким образом, чтобы находящиеся на гиперсфере нейроны никогда не могли победить и "втягивались" бы в гиперкуб учебных образцов только благодаря своим связям с ранее победившими нейронами, а не за счет собственных побед. Нетрудно оценить значение этого радиуса: максимальное расстояние между двумя произвольными точками нормированного гиперкуба учебных образцов равно $2\sqrt{d}$, следовательно, чтобы находящиеся на гиперсфере нейроны самостоятельно никогда не побеждали, должно выполняться неравенство $R > 3\sqrt{d}$. Псевдокод алгоритма выбора начальной точки на этой гиперсфере при инициализации нейронной сети имеет вид:

Переменные алгоритма:

```
X(i) // i-я координата начальной точки X на
      // гиперсфере
d     // размерность пространства учебных
      // образцов
R     // радиус гиперсферы начальных состоя-
      // ний нейронов
r     // вспомогательная переменная
```

Функции алгоритма:

```
rnd() // функция генерации случайных чисел
      // на интервале [0, 1] с равномерным
      // распределением вероятностей
iif() // условная функция, возвращающая одно
      // из двух значений в зависимости от
      // истинности проверяемого условия
```

Псевдокод алгоритма:

```
R = 10 * X() = 0
For i = 1 To d
  If i = 1 Then r = R Else r = sqrt(r^2 - X(i)^2)
  If i < d Then X(i) = r * rnd() Else X(i) = r
  X(i) = iif(rnd() < 0.5, 1, -1) * X(i)
Next
```

Собственно механизм втягивания нейронов из начальной точки на гиперсфере в нормированный гиперкуб учебных образцов и упорядоченного разворачивания нейронов сети реализован следующим образом.

Структурно-табличная модель ИНС Кохонена, обучаемая клеточным автоматом, построена так, чтобы во всех итерациях первой эпохи обучения побеждал только нейрон 0. Поэтому после первой эпохи обучения нейрон 0 и смежные ему нейроны (нейроны 1, 6, 7 на рис. 3, а) переместятся из начальной точки на гиперсфере почти в центр тяжести обучающей выборки, так как окрестностью Вороного 0-го нейрона в первой эпохе является все пространство учебных образцов. Мы говорим "почти в центр тяжести", потому что в эпохах разворачивания сети суммы приращения весов двух координат (обычно наиболее коррелированных) всех нейронов вычисляются не по формуле (1), а с небольшой мультипликативной поправкой $1 - \alpha_i/N$ и $1 - \alpha_j/N$ соответственно, где i, j — номера строки и столбца нейрона на карте Кохонена, $\alpha = 0,01...0,001$ — постоянный коэффициент, N — число нейронов в сети. Результатом действия этой поправки, учитывающей взаимное расположение нейронов на карте Кохонена, является их высокая степень самоорганизации в эпохах разворачивания сети.

После второй эпохи обучения нейрон 0 и его соседи изменяют текущее состояние в соответствии с числом своих новых побед и со структурой окрестностей Вороного, полученных в первой эпохе обучения. Кроме того, соседи нейрона 0 в свою очередь "втянут" своих ближайших соседей (нейроны 2, 8, 12, 13 и 14 на рис. 3, а) и т. д. из начальной точки гиперсферы в гиперкуб пространства учебных образцов, пока в последующих эпохах разворачивания сети все нейроны не переместятся внутрь нормированного пространства учебных образцов.

Нетрудно видеть, что для ИНС Кохонена размера $N = n \times m$ нейронов на разворачивание сети клеточным автоматом с окружением Мура затрачивается не более $\max(n, m) - 1$ эпох взаимодействия.

Если нейронная сеть не является односвязной (рис. 3, в), то ее начальная инициализация отличается от описанной, а именно: из каждой связной области необходимо поместить по одному нейрону внутрь гиперкуба учебных образцов, чтобы в первой эпохе обучения они могли самостоятельно победить; остальные нейроны помещаются в какую-либо точку на гиперсфере начальных состояний. Если этого не сделать, то после разворачивания сети активированными окажутся только нейроны связной области, к которой принадлежит нейрон 0.

Интерпретация и иллюстрация полученных результатов

Примеры разворачивания и самоорганизации нейронной сети Кохонена, обучаемой клеточным автоматом с окружением Мура (рис. 3, а), показаны на рис. 4. Как и ожидалось, конечное расположение нейронов в конце эпох взаимодействия правильно отслеживает пространственную структуру обучающей выборки данных.

Физические закономерности процесса обучения ИНС Кохонена клеточным автоматом сводятся к следующему:

1. В общем случае обучение нейронной сети выполняется за две фазы:

- фаза взаимодействия, в которой связи между нейронами включены, и сеть обучается клеточным автоматом заданной структуры;
- связи между нейронами выключены, сеть "дообучается" по алгоритму WTA.

2. В зависимости от начальных условий и структуры обучающего клеточного автомата наблюдается одна из двух устойчивых конфигураций нейронов сети в конце фазы взаимодействия — *eventually fixed* или *eventually periodic*. Очевидно, что для однозначности результата обучения сети наибольший интерес представляют конечные конфигурации *eventually fixed*. В ряде случаев вместо конфигурации *eventually periodic* удается получить конечную конфигурацию *eventually fixed* повторным обучением сети при других начальных условиях.

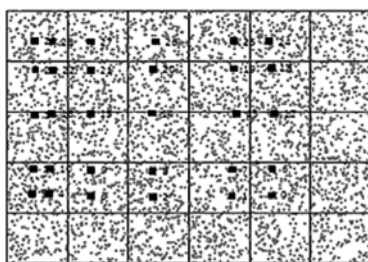
3. В конце фазы взаимодействия наблюдаются конечные устойчивые конфигурации *eventually periodic*, избавиться от которых повторным обучением сети не удастся. Типичной ситуацией, при которой такие периодические конфигурации возникают, является наличие в пространстве учебных образцов линейно-разделимых областей.

4. Длительность второй фазы обучения сети по алгоритму WTA выбирается такой, чтобы нейроны сети достигли своего стационарного состояния, которое всегда достижимо. В некоторых непакетных алгоритмах обучения ИНС Кохонена эту вторую фазу обучения называют "тонкой подстройкой" нейронов сети. При обучении нейронной сети клеточным автоматом данную фазу обучения отнюдь нельзя назвать "тонкой под-

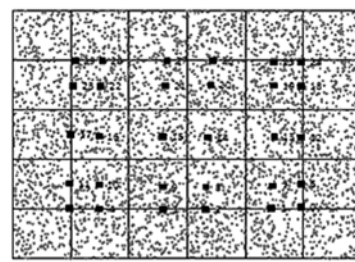
стройкой" — на деле состояния нейронов сети во второй фазе могут претерпевать существенные изменения. Так, при обучении сети Кохонена клеточным автоматом размером 2×2 с окружением Мура в фазе взаимодействия все ее нейроны сразу же смещаются в центр тяжести пространства учебных образцов и находятся там до конца этой фазы. Во второй фазе обучения (по алгоритму WTA) они "разбегаются" из центра тяжести обучающей выборки данных в свои конечные состояния на значительные расстояния.

5. "Тонкая подстройка" нейронов, наблюдаемая в непакетных алгоритмах обучения, обычно связана не с тем, что сеть уже почти обучилась и требуется лишь тонкая настройка ее нейронов, а с малой ско-

Случайная двумерная выборка данных объемом 4000 образцов

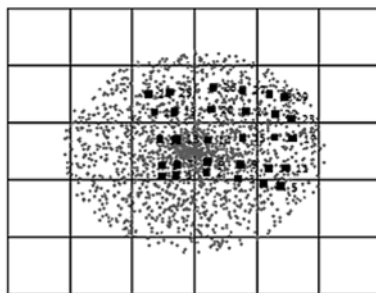


5 эпох разворачивания сети

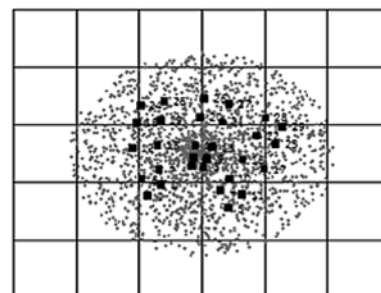


20 эпох взаимодействия

Случайная двумерная выборка с круговой симметрией данных объемом 2000 образцов

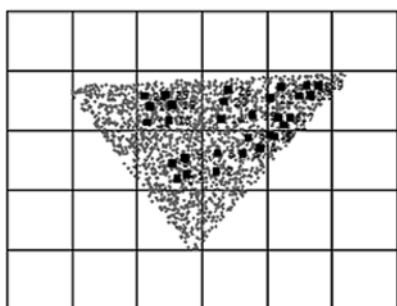


5 эпох разворачивания сети

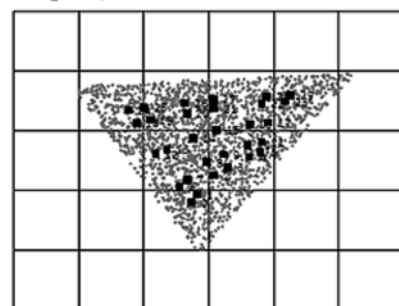


65 эпох взаимодействия

Случайная двумерная выборка с треугольной структурой данных объемом 2000 образцов



5 эпох разворачивания сети



65 эпох взаимодействия

Рис. 4. Разворачивание и самоорганизация ИНС Кохонена размером 5×6 нейронов, обучаемой клеточным автоматом с окружением Мура

ростью обучения в завершающих итерациях обучения. Действительно, постепенное замедление скорости обучения сети (в 50...100 раз) в непакетных алгоритмах обучения, с одной стороны, является принципиально необходимым, с другой стороны, противоречит здравому смыслу: чем дольше длится обучение, тем менее интенсивным оно становится. Именно поэтому обучение ИНС Кохонена непакетными алгоритмами занимает много времени и на обучающих выборках большого объема может длиться часами. Обучение сети Кохонена клеточным автоматом всегда выполняется на максимальной скорости, результатом чего является сокращение времени обучения в несколько десятков раз.

Авторы работы [10] на экспериментальных выборках данных установили ряд достоинств и недостатков пакетных алгоритмов обучения ИНС Кохонена:

- *достоинства*: простота вычислений, временная эффективность, меньшее конечное рассеяние (distortion), малое число настраиваемых параметров, детерминированный результат обучения;
- *недостатки*: плохая самоорганизация, несбалансированные кластеры, сильная зависимость от инициализации.

По мнению авторов работы [10], сильная зависимость пакетных алгоритмов обучения от инициализации имеет положительный и отрицательный аспекты:

- обеспечивает достижение лучшего локального минимума функции рассеяния;
- приводит к плохой самоорганизации.

Предложенный нами механизм начальной инициализации ИНС Кохонена, обучаемой клеточным автоматом, устраняет этот последний, самый главный недостаток пакетных алгоритмов обучения, из-за которого такие алгоритмы используются редко.

В подтверждение сказанному на рис. 5 приведены полученные нами результаты обучения ИНС Кохонена линейным клеточным автоматом размером 18×1 нейронов (см. рис. 3, б) на двумерной обучающей W -образной выборке данных.

Видим, что при использовании алгоритма обучения клеточным автоматом и предложенного меха-

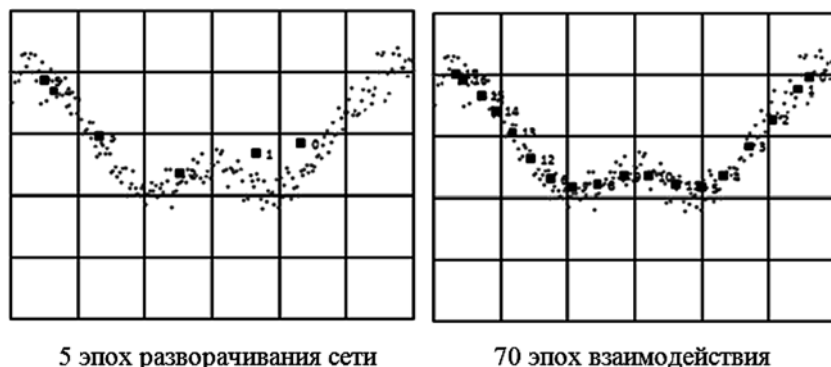


Рис. 5. Самоорганизация нейронов сети на двумерной W -образной обучающей выборке данных

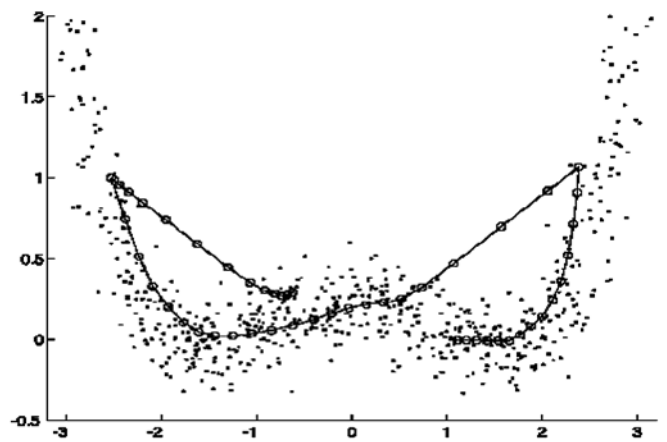


Рис. 6. Самоорганизация нейронов сети на двумерной W -образной обучающей выборке данных в работе [10]

низма начальной инициализации сети Кохонена в фазе взаимодействия наблюдается прекрасная самоорганизация нейронов, в отличие от аналогичных результатов (рис. 6), полученных авторами работы [10].

Возможность выборочного управления индивидуальными связями между нейронами ИНС Кохонена, обучаемой клеточным автоматом, может быть использована и для решения проблемы "мертвых" нейронов, т. е. нейронов, которые при обучении сети никогда не побеждают. Как отмечалось, одним из условий, приводящих к появлению "мертвых" нейронов, является линейная разделимость пространства учебных образцов.

Покажем, как избежать проблемы "мертвых" нейронов на примере решения известной задачи Фишера о классификации цветков ириса в результате использования многосвязной самоорганизующейся карты Кохонена.

В задаче Фишера рассматриваются три класса цветков ириса: Setosa (класс 1), Versicolor (класс 2) и Virginica (класс 3). Для каждого цветка ириса измеряются четыре параметра: длина и ширина чашелистника, длина и ширина лепестка. Параметры ирисов являются независимыми переменными, класс ириса — зависимой переменной. Цель состоит в том, чтобы по данным таких измерений научиться предсказывать класс нового, неизвестного цветка ириса.

Нормированная четырехмерная обучающая выборка данных содержала по 50 образцов цветков ириса классов 1, 2 и 3, где образцы класса 1 линейно отделимы от образцов классов 2 и 3. ИНС Кохонена обучалась трехсвязным клеточным автоматом, показанным на рис. 3, в, с параметрами: число прогонов — 30, эпох разворачивания сети — 5, эпох взаимодействия — 25.

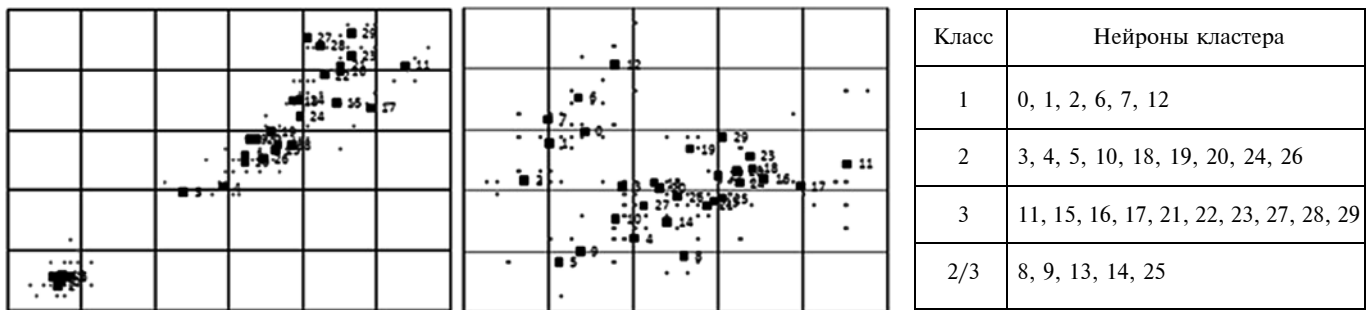


Рис. 7. Классификация цветков ириса с помощью ИНС Кохонена, обучаемой многосвязным клеточным автоматом

Начальные координаты нейронов, задаваемые при инициализации сети:

- нейрон 0 — $(-0.55, 0.08, -0.78, -0.83)$;
- нейрон 3 — $(0.20, -0.30, 0.20, 0.5)$;
- нейрон 15 — $(0.15, -0.30, 0.50, 0.68)$.

Остальные нейроны помещались на гиперсфере начальных состояний в точке с координатами $(0, 0, 9.33, 3.60)$. Отметим, что координаты нейронов 0, 3, 15 совпадают здесь с координатами одного из учебных образцов классов 1, 2 и 3 соответственно, что, как отмечалось, необходимо для успешной активации всех нейронов сети в фазе ее разворачивания.

Результаты классификации цветков ириса нейронной сетью Кохонена показаны на рис. 7, где дополнительный класс 2/3 образуют нейроны сети, расположенные на границе между линейно неотделимыми кассами 2 и 3. Очевидно, что для более точного разделения классов 2 и 3 нейронную сеть Кохонена следует обучать клеточным автоматом большей размерности.

Заключение

Проведенные экспериментальные исследования пакетного алгоритма обучения нейронной сети Кохонена клеточным автоматом убедительно показывают высокую гибкость и временную эффективность этого алгоритма. Исследования проводились с использованием разработанной авторами итерационной табличной модели ИНС Кохонена, реализующей этот алгоритм в MS Excel чисто табличными средствами, без написания программного кода VBA.

Предложен оригинальный метод начальной инициализации и адаптивного разворачивания нейронов сети, предотвращающий скручивание самоорганизующейся карты Кохонена в итерациях обучения. На тестовых выборках данных, приведенных на рис. 4, 5, опробовано множество разных способов

начальной инициализации сети, и ни один из них не показал такую высокую степень самоорганизации, какую продемонстрировал предложенный в статье метод саморазворачивания нейронной сети.

Показано, что известная проблема "мертвых" нейронов легко решается путем использования для обучения нейронной сети Кохонена многосвязного клеточного автомата.

Список литературы

1. **Кохонен Т.** Самоорганизующиеся карты: Пер. с англ. Изд. 3. М.: БИНОМ. Лаборатория знаний, 2008. С. 159—337.
2. **Lukashevich M. M., Sadykhov R. Kh.** Texture Clustering of Satellite Images Using Self-organizing Neural Network // Computing. 2008. V. 7, N. 3. P. 15—21.
3. **Shah-Hosseini H., Safabakhsh R.** A TASOM-based algorithm for active contour modeling // Pattern Recognition Letters. 2003. N. 24. P. 1361—1373.
4. **Аникин В. И., Тураева А. А.** О возможности обучения искусственной нейронной сети Кохонена с помощью клеточного автомата // Вектор науки ТГУ. 2011. № 3(17). С. 22—24.
5. **Brause R. W.** Sensor Encoding Using Lateral Inhibited Self-Organized Cellular Neural Networks // Neural Networks. 1996. V. 9. N. 1. P. 99—120.
6. **Kaneko K.** Period doubling of kink-antikink patterns, quasiperiodicity in antiferro-like structures and spatial intermittency in coupled map lattices // Prog. Theor. Phys. 1984. V. 72. P. 480—486.
7. **Аникин В. И., Аникина О. В.** Визуальное табличное моделирование клеточных автоматов в Microsoft Excel. Тольятти: Изд-во ПВГУС, 2013. С. 7—81.
8. **Wuensche A.** Classifying Cellular Automata Automatically: Finding Gliders, Filtering, and Relating Space-time Patterns, Attractor Basins, and the Z parameter // Complexity. 1999. V. 4, N. 3. P. 47—66.
9. **Аникин В. И., Аникина О. В.** Табличное моделирование клеточных автоматов в Microsoft Excel // Информационные технологии. 2009. № 6. С. 23—28.
10. **Fort J. C., Letremy P., Cottrell M.** Advantages and drawbacks of the Batch Kohonen algorithm // Proc. of 10th ESANN, Bruges, Belgium, Edition D. 2002. P. 223—230.

Learning of Kohonen Artificial Neural Network by Help of Cellular Automaton

The paper presents the systematic experimental investigation of authors' algorithm of Kohonen artificial neural network (ANN) learning by help of a cellular automaton, discrete in space and time, but continuous over the states. This kind of continuous cellular automata, introduced into use K. Kaneko in 1984, were called coupled map lattices.

An important feature of our study is that Kohonen ANN its learning algorithm have been implemented in MS Excel spreadsheet, without programming in VBA. User interface of this spreadsheet model makes it easy to change the configurable parameters and visually observe the neural network learning process.

The physical laws governing the process of Kohonen ANN learning by help of a cellular automaton are discussed.

An original method for initialization of neural network, preventing twisting Kohonen self-organizing map in learning iterations, is proposed. Examples of neural deployments and self-organization in Kohonen neural network, trained on a various number of data samples, are presented.

The possibility of solving a well-known problem of "dead" neurons using ANN cellular automaton learning and multiply Kohonen maps is demonstrated.

Keywords: neural network, Kohonen, cellular automata, learning algorithm, Excel

References

1. **Kohonen T.** Samoorganizujuschieciya karti: Per. s angl. Izd. 3. M.: BINOM. Laboratorija znaniy, 2008. P. 159—337.
2. **Lukashevich M. M., Sadykhov R. Kh.** Texture Clustering of Satellite Images Using Self-organizing Neural Network. *Computing*, 2008. V. 7. N. 3. P. 15—21.
3. **Shah-Hosseini H., Safabakhsh R.** A TASOM-based algorithm for active contour modeling. *Pattern Recognition Letters*. 2003. N. 24. P. 1361—1373.
4. **Anikin V. I., Turaeva A. A.** O vozmozhnosti obucheniya iskusstvennoi neironnoi seti Kohonena s pomoshchju kletochnogo avtomata. *Vector nauki TGU*. 2011. N. 3(17). P. 22—24.
5. **Brause R. W.** Sensor Encoding Using Lateral Inhibited Self-organized Cellular Neural Networks. *Neural Networks*. 1996. V. 9. N. 1. P. 99—120.
6. **Kaneko K.** Period doubling of kink-antikink patterns, quasi-periodicity in antiferro-like structures and spatial intermittency in coupled map lattices. *Prog. Theor. Phys.* 1984. V. 72. P. 480—486.
7. **Anikin V. I., Anikina O. V.** Vizualnoe tablichnoe modelirovanie kletochnih avtomatov v Microsoft Excel: monografiya. Togliatti: PVGUS, 2013. P. 7—81.
8. **Wuensche A.** Classifying Cellular Automata Automatically: Finding Gliders, Filtering, and Relating Space-time Patterns, Attractor Basins, and the Z parameter. *Complexity*. 1999. V. 4. N. 3. P. 47—66.
9. **Anikin V. I., Anikina O. V.** Tablichnoe modelirovanie kletochnih avtomatov v Microsoft Excel. *Informatsionnie tehnologii*. 2009. N. 6. P. 23—28.
10. **Fort J.-C., Letremy P., Cottrell M.** Advantages and drawbacks of the Batch Kohonen algorithm. *Proc. of 10th ESANN*, Bruges, Belgium, Edition D. 2002. P. 223—230.

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.В. Конова*.

Корректор *Е.В. Комиссарова*.

Сдано в набор 02.09.2014. Подписано в печать 22.10.2014. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ1114. Цена договорная.

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

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.