Intellectual Technologies on Transport No 2

Интеллектуальные технологии на транспорте № 2

Санкт-Петербург
St. Petersburg
2018

Интеллектуальные технологии на транспорте
№ 2, 2018

Сетевой электронный научный журнал, свободно распространяемый через Интернет
Публикуются статьи на русском и английском языках с результатами исследований и практических достижений в области интеллектуальных технологий и сопутствующих им научных исследований

Журнал основан в 2015 году

Учредитель и издатель

Федеральное государственное бюджетное образовательное учреждение высшего образования «Петербургский государственный университет путей сообщения Императора Александра I» (ФГБОУ ВО ПГУПС)

Адрес редакции
190031 Санкт-Петербург, Московский пр., 9, ПГУПС
email: itt-pgups@yandex.ru, сайт: http://itt-pgups.ru

ISSN 2413-2527

Журнал зарегистрирован Федеральной службой по надзору в сфере связи и массовых коммуникаций, свидетельство ЭЛ № ФС77-61707 от 07 мая 2015 г.

Журнал зарегистрирован в Российском индексе научного цитирования (РИНЦ)

Intellectual Technologies on Transport
Issue № 2, 2018

Network electronic scientific journal, open access. It publishes articles in Russian and English with the results of research and practical achievements in the field of intelligent technologies and associated research

Founded in 2015

Founder and Publisher

Federal State Educational Institution of Higher Education
«Emperor Alexander I Petersburg State Transport University»

Editorial adress
190031, St. Petersburg, Moskovskiy pr., 9, 2–108
email: itt-pgups@yandex.ru, http://itt-pgups.ru

ISSN 2413-2527

The journal is registered by the Federal Service for Supervision of Communications and Mass Media, EL no. FS77-61707 testimony from May 7, 2015

The journal is registered in the Russian Science Citation Index (RSCI)

Содержание

Contents

Application of the OpenFlow Protocol Based on the Mininet Network Emulator with the Installation of a Floodlight Controller

V.V. Jakovlev, Zh.M. Berkinbayeva
Emperor Alexander I St. Petersburg
State Transport University
St. Petersburg, Russia
jakovlev@pgups.ru, berkinbayeva.zhanniyet@gmail.com

Angel Fernandez del Campo
Universidad Politécnica de Madrid
Madrid, Spain
afc@dit.upm.es

Abstract. **OpenFlow provides accurate traffic management across the entire spectrum of switches and routers in the corporate environment, both physical and virtual, regardless of vendor. This eliminates the need to individually configure the device of each vendor through its own interface. Mininet is a network emulation platform that creates OpenFlow infrastructure elements on a single computer (physical or virtual): controller, switches, nodes, and connections. This article describes the architecture of the OpenFlow protocol, the message in the protocol, the flow table, the basic modules in the Floodlight controller architecture, and, of course, the vulnerability of the OpenFlow protocol. Results of experiment with checking the three main types of network topology with transmission of the traffic by installing the Mininet network emulator and configuring the Floodlight controller are considered.**

Keywords: **software-defined networking, OpenFlow, Floodlight controller, flow table, Mininet, Network virtualization, Quality of Service, Programmable Networks.**

INTRODUCTION

OpenFlow is an open standard that allows you to work with experimental protocols within existing networks. This standard is an extension for commercial routers, switches, and even domestic routers [1], and its implementation does not require any action on the part of the network device provider. The standard is a fundamental element of the Software Definition Network (SDN) concept.

Today, most networks do not have a high level of organization, which prevents the network from being managed effectively as required by modern requests. The OpenFlow standard solves these problems. It acts as a management protocol between network devices. In simple terms, the concept of networking SDN separates the layer of data transmission from the control layer [2]. On the basis of this, it turns out that at the data level, the simplest switches control the flow of data based on predetermined rules that are set by the central network controllers. The network structure of SDN has a high degree of virtualization [1], all ports, both virtual and physical, are managed in the same way.

GENERAL ARCHITECTURE

The first concept of OpenFlow standard was developed at Stanford University in 2008 [1]. More than a year later, in December 2009, the first version of the OpenFlow protocol was released. After the release of OpenFlow protocol was managed by the Open Networking Foundation (ONF). Shortly after its release, most companies have announced support for OpenFlow protocol in their devices. It is worth noting that OpenFlow is not the only such protocol, there are also others: OpFlex, Yang and NetConf, but they are not as common as the OpenFlow [3].

Despite the age, OpenFlow it is quite a promising development in the field of network technologies. For this reason, there is a question – is this technology necessary whether in our time. In classic routers or switches, packet transmission and high-level routing are performed on one device, whereas when using OpenFlow, these actions are separated [4]. The SDN controller in the concept of SDN networks is the brain of the entire network, transmitting data to the switches and routers "on the bottom", through southboundAPIs, and to software applications "on top", through northboundAPIs. For the organization of the work of several SDN networks, controllers must be interconnected with each other, for this purpose the OpenFlow protocol is used.

The system consists of two components: a switch with Openflow protocol and controller. Openflow switch, in turn, consists of three parts:

1) The Flow Table, which defines the actions for each flow switch

2) Secure Channel, which provides service information transfer between the controller and switch

3) The Openflow protocol, which open and standardized method of communication the switch with the controller itself. Thus, given a standard method for programming switches without having to configure each switch individually. The OpenFlow architecture is shown in figure 1.

OpenFlow is widely implemented by network equipment manufacturers because of the simple (and hence cheaper in the realization) structure of the OpenFlow switch, which can be implemented through small modifications of software and hardware. As a result, the transition to the OpenFlow protocol it is relatively easy and can be carried out step by step with the implementation of the protocol in those network segments that require OpenFlow functions.

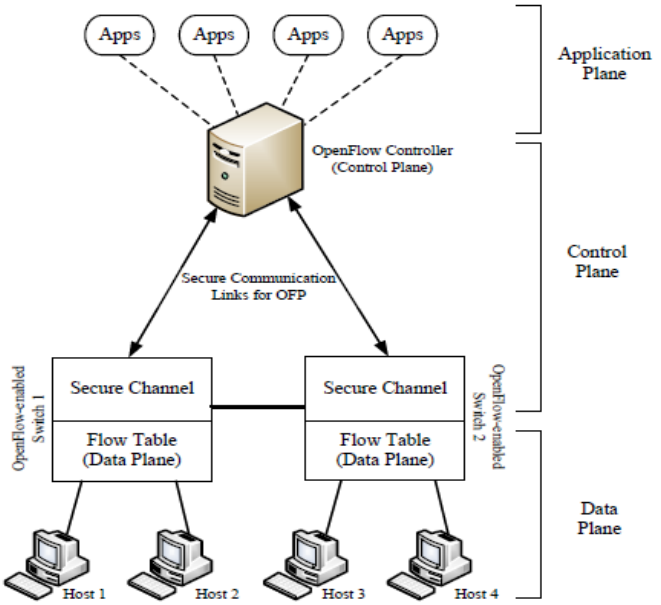


Fig. 1. Architecture of OpenFlow

MESSAGES OF OPENFLOW PROTOCOL

OpenFlow protocol can support the following types of messages: the first, controller-to-switch – used by the controller for the direct control and management of the switch state; The second type, asynchronous messages are initiated by the switch and the controller are used for notification of network events (errors, failures) and changes the switch state; third symmetrical messages can be used as the controller and the switch.

Controller-to-switch massage. The controller initiates the following message [5]:

• Features. Controller requests the status of the switch using the request features; the switch must respond using the features of the response, which indicate the possibility of a switch. This is usually done when creating OpenFlow channel.

• Configuration: the controller sets and requests the switch configuration parameters. The switch only responds to requests from the controller.

• Modify-State. Sent by the controller to manage the switch. To add / remove modify rules in the OpenFlow tables and installation the characteristics (parameters) of the switch port.

• Read-State. Used by the controller to collect statistical data from the switch.

• Packet-out. Used by the controller to send packets from a specific port on the switch and sending packets received using the Packet-in message. Packet-out messages must contain the whole packet or an identifier buffer ID, references the package that you downloaded to the switch. The massage must contain a list of actions that should be applied in this order: if the action list is empty, then the packet is reset.

• Barrier. Request / response is used by the controller to ensure that the installed dependencies between messages or to receive notifications on completed transactions. Used when necessary processing of messages in a specific order.

• Role-Request. Used to change the current role of the controller on the switch (to increase the role from Slave to Master) (for protocol version 1.3).

• Asynchronous-Configuration: this message can be used by the controller to set the filter on asynchronous messages received from the switch (for protocol version 1.3).

Asynchronous messages. Switches send asynchronous messages to the controller for notification of the arrival of the packet, change of status of the switch or error.

The following types of asynchronous messages are:

• Packet-in. If there are no corresponding rules in the switch table for the package, the switch generates a Packet-in message and sends it to the controller. For all packets sent to the virtual port CONTROLLER, Packet-in message is sent to the controller.

• Flow-Remove. When a rule for a new flow is added to the switch by using the Flow-mod message, it is set to the time-out value. This rule must be removed in this period of time because there is not enough activity or no rule is used.

• Port status. The switch can send messages to the Port status controller when the port configuration status changes.

FLOW TABLE

In the SDN switch with OpenFlow support, only the data transfer level is implemented. Each switch has its own unique table, which it fills out only on the basis of information received from the central controller. This switch table is named flowtable because the SDN network transfers data streams rather than individual packets (the rule in the switch is set only for the first packet, and then all other packets in the stream are used). These tables classify incoming packets based on the port, MAC address, IP address, and other tools [6].

Each entry in the package is cut off by a header (a bit string of a specific length). For this bit string, the flow tables, starting from the first, are looking for a rule, that has a field of the characteristics, most closer corresponds to the packet header. When there is a match, on the packet and its header conversion performed by defining a set of instructions specified in the rule results.

Recording of the flow can prescribe forward the packet to a particular port (the normal physical port or a virtual port assigned by the switch, or reserved virtual port, set the protocol specification). Reserved virtual ports Define common forwarding actions: sending a controller, broadcasting (avalanche) distribution, forwarding without OpenFlow. Virtual ports can precisely define channel aggregation groups, tunnels, or feedback interfaces [7].

If the desired rule is not found in the first table, the packet is encapsulated and sent to the controller, which generates the appropriate rule for the packets of that type and installs it on the switch (or on the set of switches it manages) or the package can be changed or reset. The processing pipeline statements allow you to forward packets to subsequent tables for further processing and to pass information between tables as metadata. The instructions also define the rules for modifying counters that can be used to collect a variety of statistics figure 2.

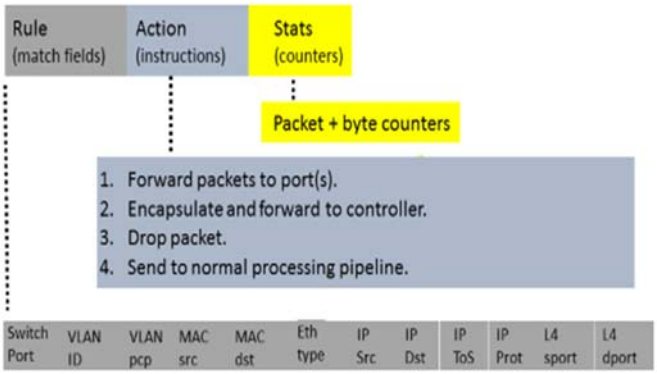


Fig. 2. Flow table in OpenFlow switch

OPENFLOW CONTROLLER

The OpenFlow controller is a kind of SDN controller that uses the OpenFlow protocol. Actually, SDN controller is the main point in SDN networks figure6. OpenFlow controller uses the same name protocol for connecting and configuring of network devices [8], to determine the best path through which the application traffic will pass. SDN controllers simplify network management by focusing the entire process of communication between applications and devices for efficient network management and modification (If, of course, it is required). Due to the fact that network management is carried out programmatically, administrators can work with traffic more efficiently, thereby increasing network performance. Summarizing, we can say that the OpenFlow controllers create a central point for managing compatible devices in the network. This protocol was created to increase the flexibility of working with the network, through the universalization of all network devices (figure 3).

Action	
1	Sends the package to all ports except inbound
2	Sends the package to the controller
3	Sends a package to the processor
4	Working with the rule table
5	Direction of the packet to the sender's port
6	Direction of the packet to the destination port
7	Resetting the package

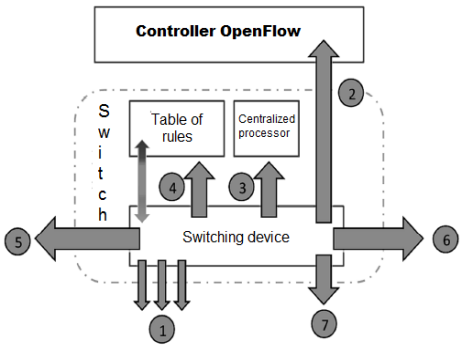


Fig. 3. Possible actions in SDN networks

VULNERABILITIES IN THE OPENFLOW POTOCOL

The advantages and disadvantages of the SDN architecture for Network infrastructure security are well understood. However, the assessment of architectural vulnerabilities should be based not only on speculation about the theoretical architecture, but also on the experiments and results of the implementation of the OpenFlow protocol in the industrial networks. The following types are currently defined threats to OpenFlow networks. Controller modes (reactive or proactive) can easily be identified by an attacker without using specific approaches or software. The identity is based on the first packet delay for the new traffic flow and is available to each user connected to the network or using the services of the infrastructure from external networks. As a result, some attacks can use the specific behavior of the controller [9]. For example, an unauthorized installation of switch processing rules that reduces the efficiency or disruption of a network is easier to implement in reactive mode due to the particularity of the Controller approach to table management flows in this mode. However, it is possible to perform this attack on the reactive controller, but it is more difficult because a complex attack is required and the likelihood of a quick attack is significantly higher. Security threats that are relevant to most information systems, such as port scanning and network service definitions, are critical to the SDN architecture because of the vulnerability of the OpenFlow channel and the large number of management traffic that is transferred between switches and network controllers [10].

CONTROLLER FLOODLIGHT

FloodLight – OpenFlow Java-controller for companies and enterprises (class enterprise) with open source. FloodLight appeared from the source code of Beacon. Has the license Apache – i.e. Can be used for any purpose.

Uses pure Java (OSGI is not required, supported by Eclipse, but is not required). Very easy to build and run. It is the core of Big Network Controller from Big Switch. Portability between applications and FloodLight Big Network Controller provides. Architecture and its features, the main components are considered in [2; 4; 8; 9; 11].

FloodLight has a modular architecture, due to which facilitates the process of expansion and modification. In describing the architecture, two main concepts are used: the service and module. Service is an interface that exports the state and generates events. Service consumers can get / set the state and to sign or unsubscribe to events. A lot of implementations of the same service are allowed. Each module, in turn, can use a certain set of services (dependencies) for implementation of some functionality. The module may provide respectively, zero or more services. That is, the modules export services. All FloodLight modules are written in Java. All modules have a minimal number of dependencies between them that simplifies application development.

The general architecture of FloodLight is shown in figure 4.

Features:

• The modular loading system, which makes it possible to expand and increase the functionality of the controller.

• Easy installation with minimal dependencies.

• Supports a wide range of physical and virtual OpenFlow switches.

• Supports integration with non-OpenFlow networks, ie it can manage a lot of "islands" of physical OpenFlow switches.

• One of the main development goals – high performance.

• Supports platform OpenStack cloud orchestration.

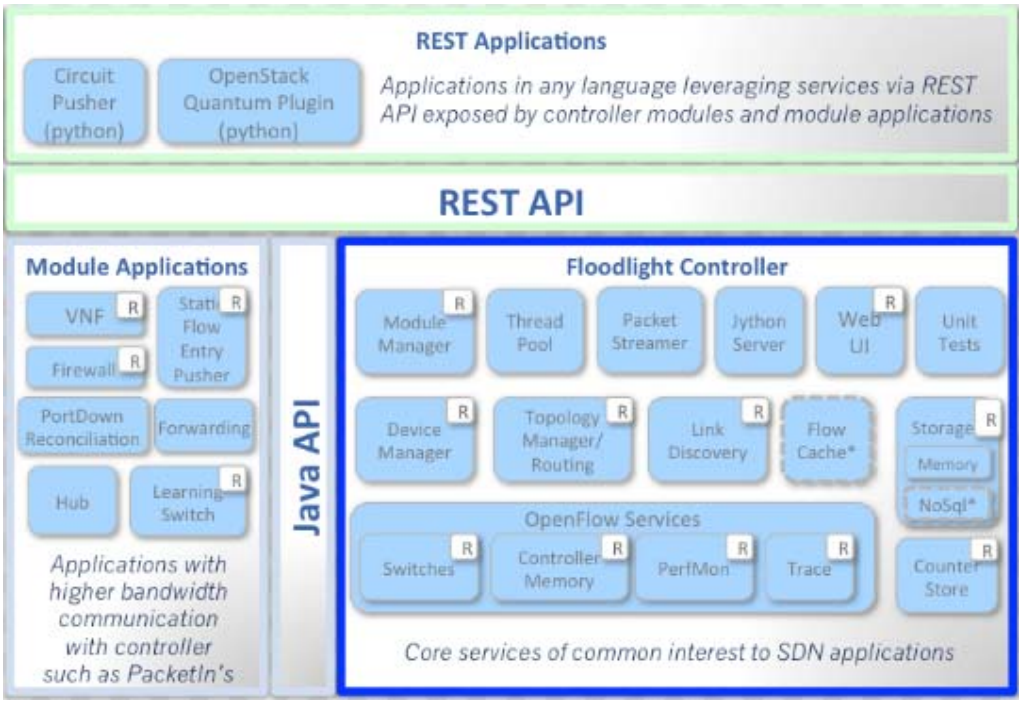


Fig. 4. Architecture and basic modules of FloodLight

FEATURES AND OPPORTUNITY OF MININET

Mininet is a computer network emulator in the Linux environment. Mininet creates virtual OpenFlow networks-controllers, switches, and hosts-on one real or virtual machine. Enables you to quickly create, communicate, and configure software-defined networks. Almost every operating system is a computational resource virtualization using process abstraction. Mininet uses a virtualization-based process to launch many hosts and switches on one core OS. The mininet can create a kernel or a custom space of openflow-switch, a controller for the fabric control, and organizes a connection on the simulated network. Mininet connects switches and hosts using virtual Ethernet pairs. Currently, Mininet supports only Linux, but in the future it can support other operating systems such as Solaris or FreeBSD.

Mininet supports five built-in network topologies, namely, a minimal topology, a single topology, a linear topology, a tree topology, and a reverse topology [12].

MININET INSTALLATION AND CHECK OF DIFFERENT NETWORK TOPOLOGIES

Starting with the 2.6.24 version, the Linux kernel supports the virtualization and isolation Mechanisms-cgroups [13], which provide network interfaces, routing tables, and ARP tables with processes within the same operating system. This is one type of OS-level virtualization that allows you to run many of the same processes in an isolated and resource-limited environment.

The default topology is the minimum topology, which is predefined with one OpenFlow kernel switch connected to two host controllers and OpenFlow, while the number of switches and nodes can be changed for other topologies using the command-line interface (CLI).

How to use it

All the work with a virtual network mininet, namely to deploy the network of the desired topology, change different host or switch settings, and so on, is done in a simple command interpreter- mn:

```
$ sudo mn
```

Run without parameters, MN will go into the command's interpretation mode. By default, it will create a minimal network consisting of two hosts (h1, h2), switch (s1), and OpenFlow Controller (c1):

```
$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet>
```

The Mininet interpreter uses a number of native commands that allow virtual network management to be almost real. The following are examples of key features. You can display a list of all hosts, switches, and controllers by using the command: nodes

To view a network topology, you can use the following command to map the switch and host ports: net

You can display the network interface configuration of a particular host using the classic ifconfig command before which you specify a specific host name:

```
mininet> h1 ifconfig
```

In the above command, you can specify a specific switch instead of the host name, and the configuration of its ports will be displayed.

You can turn off and turn on any of the switch ports as desired:

```
mininet> link s1 h1 down
mininet> link s1 h1 up
```

You can view the routing table for a particular host using the following command: route
Run Ping:

```
mininet> h1 ping h2
```

Ping everyone:

```
mininet> pingall
```

Actually, you can perform most of the standard Linux commands on each of the hosts by specifying its name first. For example, to see the processes of any of the hosts or switches, the same: ps

```
mininet> s1 ps
```

You can end any of the processes by using the standard: Kill-9. In addition to checking the availability of nodes using ping, you can still test bandwidth between nodes using the old: iperf

```
mininet> iperf h1 h2
```

Interface throughput can be limited from 10 to 1000 Mbit/s. Well, after all, you can just get the terminal to any of the nodes:

```
mininet>  xterm h1
```

Additional services.

On each of the virtual hosts, in addition to the standard processes, you can run third-party services. For example, this could be a simple Web server in Python:

```
mininet> h1 python -m SimpleHTTPServer 80
&
```

and try to connect to it from another node:

```
mininet> h2 wget -O – h1
```

You can shut down the Web server if you want to:

```
mininet> h1 kill %python
```

The mininet virtual network does not exist permanently — it is created when you write MN with or without specific parameters, and is destroyed when the interpreter exits. All this is happening almost instantaneously. Even a large network with several hundred hosts and dozens of switches are created in seconds. And it's all on a single-processor virtual machine with one gigabyte of RAM.

You can collapse the entire virtual network and exit the OS shell with the following command: quit.

If the interpreter's work was not completed correctly, you can get rid of the dangling processes and other service data by using the command:

```
$ sudo mn –c
```

Now, by complicating the topology, I will show you three main topologies of the OpenFlow network: Single, Linear and Tree topology.

There are four basic topologies that you can use, "not would" with the Python syntax — they are already described and implemented as MN parameters. Here are some details about each one.

Minimal. As has been previously shown, the default is to run the MN without parameters. In this case, you create two hosts that are connected to the same switch, which in turn is controlled by the OpenFlow controller. You cannot specify an arbitrary number of hosts or switches in this topology.

Single. As with minimal, all hosts connect to the same switch. The only difference is the ability to specify the number.

Single topology.

The Single topology consists of a single OpenFlow-enabled switch connected to multiple hosts as defined. The switch in turn connected to the OpenFlow controller though secure channel. A single topology has 16 numbers of hosts is developed in Mininet using CLI command as:

```
$ sudo mn --topo = single,16
```

When running the above command for OpenFlow-enabled single topology in command line, a Mininet console will create a single OpenFlow-enabled network topology has 16 hosts, and is connected to a single OpenFlow-enabled switch. The switch in turn is connected with control plane (an OpenFlow controller) as shown in figure 5.

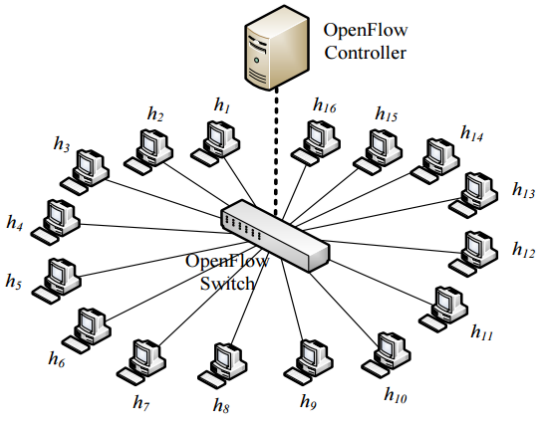


Fig. 5. OpenFlow single topology with 16 hosts

Linear topology

In linear topology, if there are 'n' hosts on the network then 'n' numbers of switches are required. Which means that each host will be connected to the appropriate switch. For example, host h1 will connect to switch S1, host h2 with switch s2, and all the switches are connected to one another which in turn is connected to a common controller. A linear topology with 16 hosts is designed in Mininet using CLI command as:

```
$ sudo mn --topo = linear,16
```

Similarly, when execution of above command for the linear topology on the command line, the Mininet console will create a linear topology OpenFlow support with 16 hosts. Since, as discussed that each host is connected to its own switch, 16 switches are also required in the network and the switches are connected with each other as shown in figure 6.

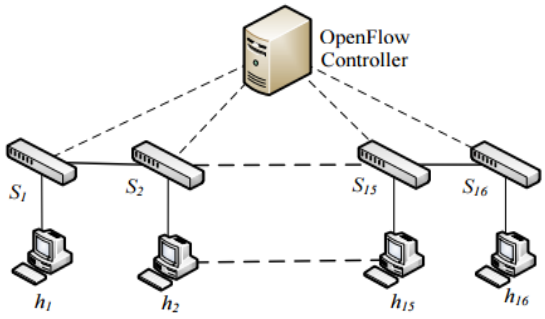


Fig. 6. OpenFlow linear topology with 16 hosts

A linear topology with 16 hosts connected to its own switches in linear fashion is clearly shown in figure 10. All switches are interconnected with each other and in turn connected with OpenFlow controller as shown in the figure 6.

Tree Topology.

A tree topology is depending on the location of the switches and hosts in a tree fashion. This means, that there are multiple branches in the topology and in these branches multiple switches and hosts are connected according to topological design. A tree topology which has 16 hosts is developed at the command line CLI using the following command:

```
$ sudo mn --topo = tree,depth=2,fanout=4
```

In the above CLI command, to create a tree topology, the syntax of the command determines the depth and fanout. Here, the depth indicates the number of levels of switches and the fanout indicates the number of available output ports to connect switches or hosts. Depth is require for the number of levels of switches to connect starting from controller. This means that, let the controller be at level '0', then according to this example there will be two levels of the switch, and finally, the level of hosts will appear. The number of hosts require to connect with each switch depends on the number of fanout, the fanout in this example is 4. In this example, the number of switch levels is two and each switch has 4 numbers of output ports for connection of next level. The above command will create a tree topology that has 16 hosts in Mininet as shown in figure 7.

The command presented above builds a network with the classic three tiered model. When each access-level switch is connected to four hosts. The access switches will in turn be included in the distribution of the four switches, which in turn are placed in a single core. Unfortunately, such capabilities as stacking switches or aggregation of channels and the standard VLANs are not implemented in Mininet, making the model not entirely realistic. But in general terms, it's quite similar.

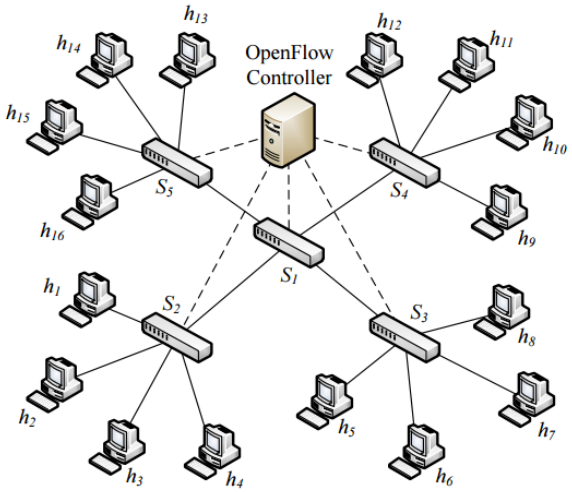


Fig. 7. OpenFlow tree topology with 16 hosts

To build a network using its own topology, you can use a key: custom

```
$ sudo mn –custom /<script path>/topo-
    2sw-2host.py –topo mytopo
```

Network settings.

By default, all objects in the Mininet network are connected by virtual gigabit channels. This can be verified by executing the IPERF command in the Mininet interpreter, which will test bandwidth between the two network objects.

```
mininet> iperf
```

INSTALLATION AND SETUP OF FLOODLIGHT CONTROLLER

Floodlight is open-source software, whose source code is published on GitHub: github.com/floodlight/floodlight. However, it is easier to use the integrated development environment (IDE), such as Eclipse, to add new modules. The next steps are necessary to download floodlight and integrate it with Eclipse.

Enter the following commands:

```
sudo apt-get install build-essential
default-jdk ant python-dev eclipse git
clone
git://github.com/floodlight/floodlight.g
it
cd floodlight
ant eclipse
```

1. Launch Eclipse

2. "File" → "Import" → "General" → "Existing Projects into Workspace" → "Next"

3. In the "Select root directory" click "Browse" and select the parent directory where you placed ßoodlight. → click "Finish"
4. Create the FloodlightLaunch target:

   (a) Click "Run" → "Run Configurations"

   (b) Right Click on "Java Application" → "New"

- For "Name" use *"FloodlightZhanniuyet"*
- For "Project" use *"Floodlight*
- For "Main" use *"net.ßoodlightcontroller.core.Main"*

   (c) Click "Apply"

Adding a module to Floodlight.
In order to add a module with new functionalities, it is needed to change the default startup modules:

```
src/main/resources/floodlightdefault.properties
src/main/resources/META-INF/services/net.floodlight.core.module.IFloodlightModule
```

Adding the new class created.

Conclusion

As a result of the work done, it has been found that the application of the OpenFlow protocol to configure the switch allows you to dynamically configure the network infrastructure components. To simulate a script by using SDN, this project used the Mininet software which developed by Stanford University and released under the BSD open source to simulate the network and floodlight as a network controller. Another tool that is used is virtual box to launch the Mininet virtual network.

By using the Mininet simulated various network topologies with traffic transfer. This approach to modelling SDN networks has the following advantages over analogues: rapid creation of network topology, portability of code in real controllers, virtual network on local PC, independence from the manufacturer of network equipment and good scalability. The creation of virtual networks based on Mininet promotes the flexible, dynamic development of network applications, allowing the creation of innovations using software.

References

1. Smelyansky R. Integrated environment for the analysis and design of distributed real-time embedded computing systems, *Programming and Computing Software*, 2012, No. 9, pp. 242–254.

2. Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R. Advanced study of SDN/OpenFlow controllers, *Proc. 9th Central & Eastern European Software Engineering Conference in Russia*, Moscow, 2013, pp. 105–110.

3. Akyildiz F., Ahyoung L., Wang P., Min L., Chou W. A roadmap for traffic engineering in SDN-OpenFlow networks. 2014. No. 71, pp. 2–30.

4. Autenrieth A., Elbers J., Kaczmarek P. Cloud orchestration with SDN/OpenFlow in carrier transport networks, *Proc. Transparent Optical Networks (ICTON), 2013 15th International Conference on. – IEEE*, Bristol, 2013, pp. 274–282.

5. Quality of Service (QoS). Available at: http://www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html.

6. ONF Specification//Open network foundation. Available at: https://www.opennetworking.org/sdn-resources/onf-specifications.

7. Multiprotocol Label Switching Traffic Engineering Technology Overview. Available at: http://www.ciscopress.com/articles/article.asp?p=520184.

8. Karimzadeh M., Valtulina L., Applying G. SDN / OpenFlow in Virtualized LTE to support distributed mobility management (DMM), *Proc. Proceedings of the 4th International Conference on Cloud Computing and Services Science, CLOSER 2014*, Portugal, 2014, pp. 639–644.

9. Diego K., Fernando R., Paulo V. Towards secure and dependable software-defined networks, *Proc. The second ACM SIGCOMM workshop on Hot topics in software defined networking, 2013*, pp. 14–76.

10. Margaret Wasserman and Sam Hartman. Security analysis of the open networking foundation (onf) openflow switch specification. Available at: https://tools.ietf.org/id/draft-mrw-sdnsec-openflow-analysis-00.html.

11. Vidya B. Harkal A., Deshmukh A. Software Defined Networking with Floodlight Controller, *Proc.International Conference on Internet of Things, Next Generation Networks and Cloud Computing*, Portugal, 2016, pp. 23–27.

12. Lantz, B., Heller, B., & McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. 2010. Available at: http://conferences.sigcomm.org/hotnets/2010/papers/a19-lantz.pdf.

13. Cgroups technology. Available at: https://ru.wikipedia.org/wiki/Cgroups.

14. Floodlight is an Open SDN Controller, Project Floodlight. Available at:http://www.projectfloodlight.org/Floodlight.

15. How to Use Access Control Lists (ACLs). Available at: http://symfony.com/doc/current/security/acl.html.

Применение протокола OpenFlow на базе эмулятора сети Mininet с установкой контроллера Floodlight

В.В. Яковлев, Ж.М. Беркинбаева
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, РФ
jakovlev@pgups.ru,
berkinbayeva.zhanniyet@gmail.com

Ангел Фернандез дел Кампо
Мадридский политехнический университет
Мадрид, Испания
afc@dit.upm.es

Аннотация. **Протокол OpenFlow обеспечивает точное управление трафиком по всему спектру коммутаторов и маршрутизаторов в корпоративной среде, как физических, так и виртуальных, независимо от поставщика. Это устраняет необходимость индивидуальной настройки устройства каждого поставщика через собственный интерфейс. Mininet является платформой эмуляции сети, которая создает на одном компьютере (физическом или виртуальном) элементы инфраструктуры OpenFlow: контроллер, коммутаторы, узлы и соединения.**

В статье описывается архитектура протокола OpenFlow, сообщение в протоколе, таблица потоков, базовые модули в архитектуре контроллера Floodlight и уязвимость протокола OpenFlow. Рассматриваются результаты эксперимента по проверке трех основных типов топологии сети с передачей трафика путем установки сетевого эмулятора Mininet и настройки контроллера Floodlight.

Ключевые слова: **программно-конфигурируемые сети, OpenFlow, контроллер Floodlight, таблица потоков, Mininet, виртуализация сети, качество обслуживания, программируемые сети.**

ЛИТЕРАТУРА

1. Smelyansky R. Integrated environment for the analysis and design of distributed real-time embedded computing systems / Smelyansky R. // *Programming and Computing Software*. – 2012. – № 9. – Pp. 242–254.

2. Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R. Advanced study of SDN/OpenFlow controllers / Shalimov A. // *Proc. 9th Central & Eastern European Software Engineering Conference in Russia*. – 2013. – Pp. 105–110.

3. Akyildiz F., Ahyoung L., Wang P., Min L., Chou W. / Akyildiz F // A roadmap for traffic engineering in SDN-OpenFlow networks. – 2014. – № 71. – Pp. 2–30.

4. Autenrieth A., Elbers J., Kaczmarek P. Cloud orchestration with SDN/OpenFlow in carrier transport networks/ Autenrieth A. // *Proc. Transparent Optical Networks (ICTON), 2013 15th International Conference on. – IEEE*. – 2013. – Pp. 274–282.

5. Quality of Service (QoS). URL: http://www.cisco.com/c/en/us/products/ios-nx-os-software/quality-of-service-qos/index.html.

6. ONF Specification//Open network foundation. URL: https://www.opennetworking.org/sdn-resources/onf-specifications.

7. Multiprotocol Label Switching Traffic Engineering Technology Overview. URL:http://www.ciscopress.com/articles/article.asp?p=520184.

8. Karimzadeh M., Valtulina L., Applying G. SDN / OpenFlow in Virtualized LTE to support distributed mobility management (DMM) / Karimzadeh M. // *Proc. Proceedings of the 4th International Conference on Cloud Computing and Services Science, CLOSER 2014*. – 2014. – Pp. 639–644.

9. Diego K., Fernando R., Paulo V. Towards secure and dependable software-defined networks / Diego K. // *Proc. The second ACM SIGCOMM workshop on Hot topics in software defined networking. – 2013*. – Pp. 14–76.

10. Margaret Wasserman and Sam Hartman. Security analysis of the open networking foundation (onf) openflow switch specification. URL: https://tools.ietf.org/id/draft-mrw-sdnsec-openflow-analysis-00.html.

11. Vidya B., Harkal A., Deshmukh A. Software Defined Networking with Floodlight Controller / Vidya B. // *Proc. International Conference on Internet of Things, Next Generation Networks and Cloud Computing*. – 2016. – Pp. 23–27.

12. Lantz, B., Heller, B., & McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. 2010. URL:http://conferences.sigcomm.org/hotnets/2010/papers/a19-lantz.pdf.

13. Cgroups technology. URL:https://ru.wikipedia.org/wiki/Cgroups.

14. Floodlight is an Open SDN Controller, Project Floodlight. URL:http://www.projectfloodlight.org/Floodlight.

15. How to Use Access Control Lists (ACLs). URL: http://symfony.com/doc/current/security/acl.html.

Перспективы внедрения автоматизированного рабочего места ведения технической документации в Петербургском метрополитене

Д.В. Зуев, Д.В. Седых, Д.В. Сомов
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, Россия
zuevdv@gmail.com, sedyhdmitriy@gmail.com, somov.sdv@yandex.ru

Аннотация. **Рассматриваются вопросы разработки комплекса средств автоматизированного создания технической документации, контроля ее правильности, проверки работоспособности проектируемых систем и организации обмена информацией, а также возможности создания современных жизнеспособных и конкурентоспособных систем электронного документооборота, позволяющих решать вопросы, связанные с потребностями транспортного комплекса по распределению и обработке информационных потоков.**

Ключевые слова: **техническая документация, электронный документооборот, автоматизация, схематический план станции, двухниточный план станции, принципиальные схемы, локальная вычислительная сеть.**

Введение

На сегодняшний день в Петербургском метрополитене функционирует 5 линий, эксплуатационная длина которых составляет 113,6 км. За сутки по подземным артериям проходят 3,5 тысячи поездов. Ежедневно услугами метро пользуются более 2 миллионов пассажиров. Действуют 67 станций (в том числе 7 пересадочных узлов), 12 из которых совмещены с вокзалами или ж.-д. станциями, 5 электродепо.

Важнейшими в этом сложном хозяйстве являются службы движения, пути, сигнализации и связи, электроснабжения, подвижного состава, тоннельных сооружений, электромеханическая и эскалаторная.

На любом современном предприятии основным способом представления информации является документ. Неоспорима важность сохранности и умелого использования информационных ресурсов предприятия. Способность принимать верное решение и вовремя реагировать на все изменения, возникающие при управлении метрополитеном, зависит не только от таланта и опыта руководителей, но и от того, насколько разумно в нем организовано управление документооборотом.

Анализ состояния документооборота технической документации в хозяйстве метрополитена показывает, что используется преимущественно электронно-бумажная технология, эффективность которой чрезвычайно низка: горы бумаги, шкафы, стеллажи, принтеры, сканеры, плоттеры, дорогостоящие комплектующие, все согласования и утверждения выполняются только на бумаге, большое количество командировок и почтовых расходов, потери рабочего времени и т. п.

Большинство технических документов, необходимых в повседневной работе, хранятся в виде отсканированных копий, актуальность которых необходимо постоянно сверять с архивом и делать новые копии при изменении документа. На это уходит много времени, что существенно сказывается на сроках реализации как новых проектов, так и проектов по реконструкции. Естественно, что при использовании бумажной технологии невозможно распараллелить процесс подписания документа и достаточно сложно контролировать его прохождение по маршруту.

Единственная возможность повысить эффективность прохождения технической документации (ТД) при согласовании и утверждении – переход на электронный документооборот (ЭД). Не забываем и об основной задаче структурных подразделений метрополитена, которая заключается в надежном обслуживании и поддержании работоспособности технических средств и объектов, а также об активном участии в проектировании новых объектов и внедрении новых технических средств. Качественное техническое обслуживание немыслимо без технической документации, которая должна быть удобна, доступна и актуальна. Для обеспечения этих требований рекомендуется внедрение на предприятии электронного документооборота технической документации.

К электронному документообороту технической документации необходимо предъявлять следующие основные требования:

– получение схематических планов станций и путевых планов перегонов проектными организациями в электронном виде в отраслевом формате технической документации (ОФТД);

– передача всех проектов, принципиальных схем и технических документов в электронном виде в ОФТД;

– согласование и утверждение технической документации с помощью электронной подписи;

– автоматизированная проверка соответствия всех стадий выполнения проекта заданию на проектирование;

– промежуточный автоматизированный контроль выполнения всех стадий проекта службами метрополитена;

– автоматизированный контроль выполнения всех стадий проекта субподрядными организациями;

– перевод бумажного архива технической документации в электронный вид с конвертацией в отраслевой формат.

ПРЕДЛОЖЕНИЯ ПО ВНЕДРЕНИЮ

Для повышения эффективности процессов ведения и использования технической документации за счет использования компьютерных технологий ее получения, хранения и переработки предлагается реализовать систему электронного документооборота технической документации в хозяйстве службы сигнализации, централизации и блокировки (служба Ш) Петербургского метрополитена на базе АРМ ВТД (рис. 1), [1–3].

К задачам внедрения АРМ-ВТД относятся: сокращение времени поиска необходимой информации; сокращение времени получения копий документации; сокращение затрат времени и повышение качества работ; сокращение затрат времени и повышение качества контроля изменений, вносимых в документацию; сокращение числа отказов устройств; экономия эксплуатационных расходов дистанции; снижение доли бумажных документов [4; 6].

Основными пользователями АРМ-ВТД являются: инженеры службы Ш, дистанций СЦБ, руководители служб и отделов, принимающих участие в согласовании ТД.

Основные функции системы

1. Хранение:
   - клиент-серверная система документооборота с резервным копированием данных;
   - хранение в базе данных любых видов документов;
   - организация хранилища с учетом структуры.
2. Мгновенный доступ к документации:
   - различные варианты удаленной работы и доступа к документации;
   - мгновенный доступ к документации, встроенная поисковая система;
   - обеспечение взаимодействия с проектными организациями на уровне безбумажной технологии.
3. Специализированные функции:
   - поддержка отраслевого формата технической документации [5; 7–11];
   - специализированные редакторы для работы с документацией;
   - поддержка электронной подписи для согласования.
4. Автоматизация работы:
   - автоматизация создания и редактирования схем;
   - передача проектной документации в базу ТД;
   - автоматизированное построение изображений объектов по параметрам;
   - формирование спецификаций по чертежу, по списку чертежей, по объекту;
   - предоставление возможности ручной коррекции информации;
   - вывод на печать, запись в файл.
5. Интеллектуальные функции:
   - модули синтеза схем [12; 13];
   - модули моделирования работы систем [14; 15];
   - модули экспертизы схем [16–18];
   - модули распознавания технической документации.
6. Обеспечение безопасности:
   - авторизация пользователей;
   - поддержка электронной подписи для согласования документации в электронном виде;
   - настройка прав доступа и режимов безопасности.

АРМ ВТД может быть частично развернут на существующей инфраструктуре, поэтому не требует значительных капитальных вложений. Локальный сетевой трафик может транспортироваться через технологическую сеть связи метрополитена (ТССМ), а удаленный доступ с мобильных устройств для нужд аварийных служб и оперативно-ремонтного персонала может осуществляться через сеть WiFi, доступную на всех станциях и перегонах. Используемые в настоящее время в метрополитене аппаратные средства в полной мере удовлетворяют системным требованиям к развертыванию АРМ ВТД.

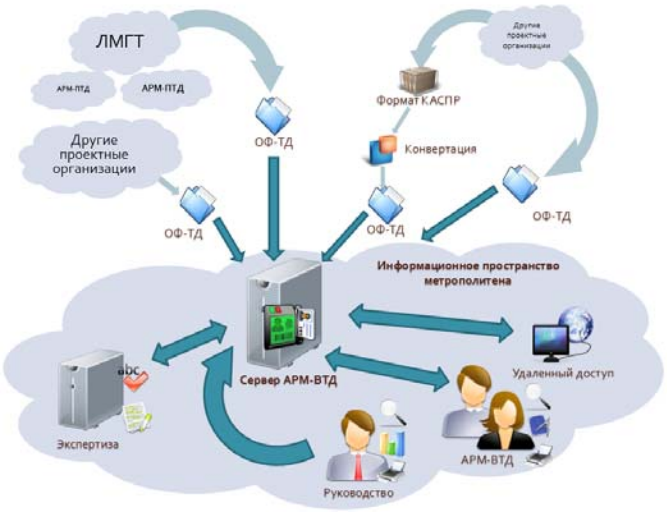


Рис. 1. Единое информационное пространство

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Для определения возможности внедрения АРМ ВТД в метрополитене необходимо исследовать структуру подразделений, организацию работы с технической документацией, типы технической документации, ответственность должностных лиц и многое другое. После чего отдельные составляющие АРМ ВТД могут быть настроены или отдельно адаптированы для работы в конкретных подразделениях метрополитена.

Рассмотрим пример внедрения АРМ-ВТД в службе сигнализации, централизации и блокировки (служба Ш).

Вся техническая документация в службе Ш ведётся согласно инструкции составленной в соответствии с действующими ГОСТами, ОСТами, строительными нормами и правилами (СНиП) и утверждённой главным инженером – первым заместителем начальника метрополитена. В инструкции установлены правила разработки, внесения изменений, дублирования, согласования и хранения технической документации на устройства СЦБ, являющиеся обязательными для всех работников службы Ш, а также других служб и организаций, занимающихся эксплуатацией, проектированием, строительством и ремонтом устройств СЦБ.

На участках, в дистанциях, в службе Ш и других службах на все действующие устройства СЦБ имеется техническая документация в соответствии с п. 6.50 ПТЭ РФ.

Основные технические документы и места их хранения в службе Ш приведены в таблице.

Вид технического документа	Экземпляр участка СЦБ	Экземпляр дистанции СЦБ	Экземпляр службы
Схематические планы станций	V	V	V
Таблицы взаимозависимости стрелок, сигналов, маршрутов	V	V	V
Двухниточные планы станций, перегонов и парковых путей, которые могут быть совмещены с кабельными планами	V	V	V
Внешний вид аппаратов управления и табло	V		
Комплект принципиальных и монтажных схем	V		
Схемы электроснабжения устройств СЦБ	V		
Инструкция по техническому обслуживанию устройств СЦБ и сборник технологических карт со всеми дополнениями и изменениями	V	V	
Действующие инструкции, приказы, указания, связанные с обслуживанием устройств	V	V	V
Инструкция о порядке пользования устройствами МРЦ (БМРЦ, МРЦ с АРМ ДСЦП), станционной аппаратуры комплексной системы обеспечения безопасности движения и автоматизированного управления поездов (СА КСД)	V	V	V
Исполненный кабельный план	V	V	
Справочный материал на аппаратуру	V	V	
Графоаналитические расчеты пропускной способности перегонов, станций и соединительных ветвей		V	V
Кабельные планы		V	V
Чертежи на установку напольного оборудования СЦБ		V	
Схемы трасс магистральных кабелей		V	V
ТУ, ТО, типовые проектные решения, учебная литература		V	V

На диспетчерском участке (ДУ) имеется следующая техническая документация по обслуживанию устройств диспетчерской централизации (ДЦ) и комплексной автоматизированной системы диспетчерского управления (КАСДУ):

– на диспетчерском посту (ДП): комплекты принципиальных и монтажных схем, технические описания систем ДЦ (СКЦ-67), ДЦМ и КАСДУ, инструкции о порядке пользования устройствами диспетчерской централизации линий, инструкции о порядке пользования устройствами МРЦ (БМРЦ), инструкции о порядке пользования устройствами МРЦ с АРМ ДСЦТ станций, инструкция по инсталляции программного обеспечения, однолинейные и принципиальные схемы электропитания КАСДУ;

– на линейном посту (ЛП): комплект принципиальных и монтажных схем, инструкции по пользованию АРМ электромеханика, однолинейные и принципиальные схемы электропитания КАСДУ.

В дополнение к основной документации при строительстве и реконструкции устройств СЦБ служба управления ремонтами и эксплуатацией передает в службу Ш четыре экземпляра проекта (при децентрализованном размещении оборудования – пять) с учетом передачи не менее двух экземпляров до начала производства работ. В дистанции ремонта, участке надежности и контроля службы имеются паспорта, чертежи, технические требования, стандарты и другие нормативные документы на оборудование и устройства СЦБ.

За ведение технической документации на участках отвечает старший электромеханик (начальник участка), на уровне дистанций – инженер (ведущий инженер), на уровне службы – главный инженер.

Все перечисленные выше схемы, планы, чертежи могут быть нарисованы средствами АРМ, а отображение и параметры доступа могут быть настроены согласно организационной структуре службы Ш. Структурная схема программного комплекса АРМ приведена на рис. 2.

Описание основных подсистем

1. Базовые задачи (модуль БЗ). Данная подсистема обеспечивает автоматизацию формирования и ведения пользователей системы аппарата управления, служб и других подразделений, реализует механизм разграничения прав доступа, обеспечивает взаимодействие серверной и клиентской частей ПО, администрирование и создание оболочки программного комплекса, которая отвечает за формирование «дерева объектов» и технических документов. В задачи подсистемы входит сохранение чертежей проекта в открытом отраслевом формате. ОФТД реализует возможность передачи чертежей и схемных решений в другие информационные системы для просмотра и анализа.

2. Задачи ведения и делопроизводства (модуль ЗВ). Данная подсистема обеспечивает решение взаимосвязанных задач организации работы и выполнение технологических операций с нормативно-правовыми документами Российской Федерации, оперативно-распорядительной документацией (ОРД) метрополитена. Реализует оперативный процесс принятия управленческих решений, таких как согласование и утверждение ТД, а также обеспечивает возможность внесения изменений в документацию. Структурная схема модуля ЗВ отображена на рис. 3.

3. Средства рисования (модуль СР) представляют собой подсистему, состоящую из универсального графического редактора, набора специализированных библиотек элементов схем связи и специализированных библиотек для дальнейшей разработки модулей автоматизации. Посредством универсального графического редактора данная подсистема осуществляет поддержку цикла разработки чертежей и схемных решений. Автоматизация двухмерного графического редактора обеспечивает реализацию функций, соответствующих определенному типу чертежа и специфике работы подразделения. Структурная схема модуля СР отображена на рис. 4.

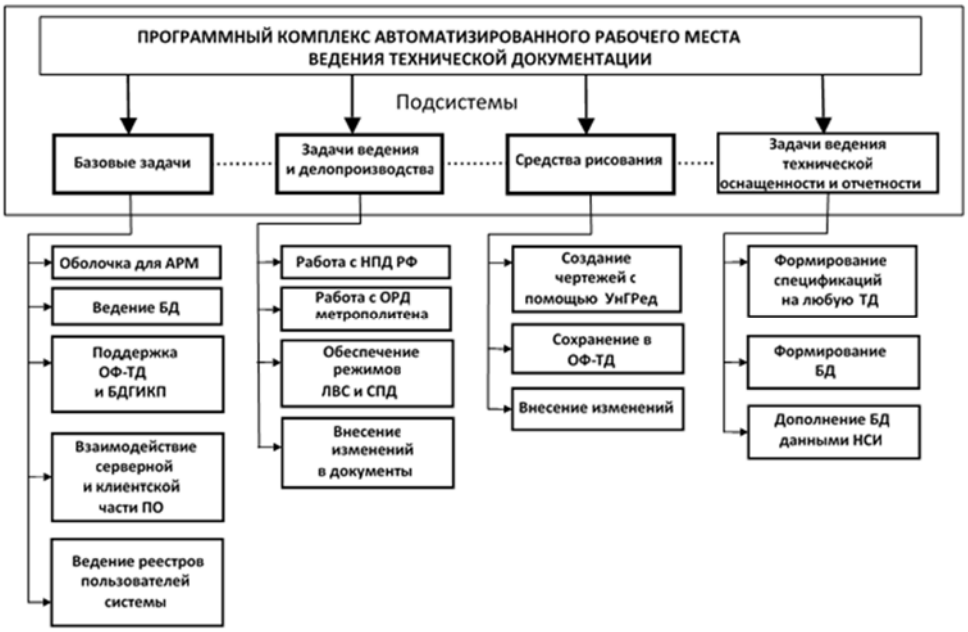


Рис. 2. Структурная схема АРМ ВТД

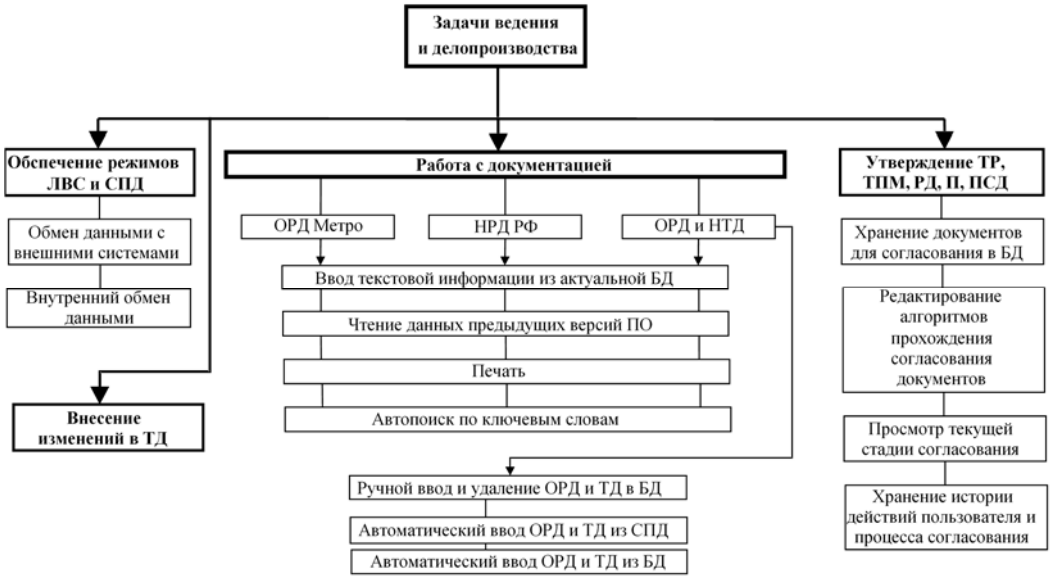


Рис. 3. Структурная схема модуля ЗВ

4. Задачи ведения технической оснащенности и отчетности (модуль ВО). Основной задачей является автоматизированное формирование паспортов объектов связи и заказных спецификаций на любую ТД. Подсистема выполняет следующие функции:

– формирование и ведение данных по оснащенности устройств;
– формирование и корректировка баз данных по оборудованию;
– формирование и ведение баз данных «Паспорта объектов».

Подсистема выполняет анализ технической оснащенности устройств, в задачи которого входит реализации следующих функций:

– расчет и автоматизированное формирование паспортов объектов;
– формирование ведомостей оснащенности устройствами;
– формирование схемы технической оснащенности подразделения;

получение справочно-аналитических форм, диаграмм.

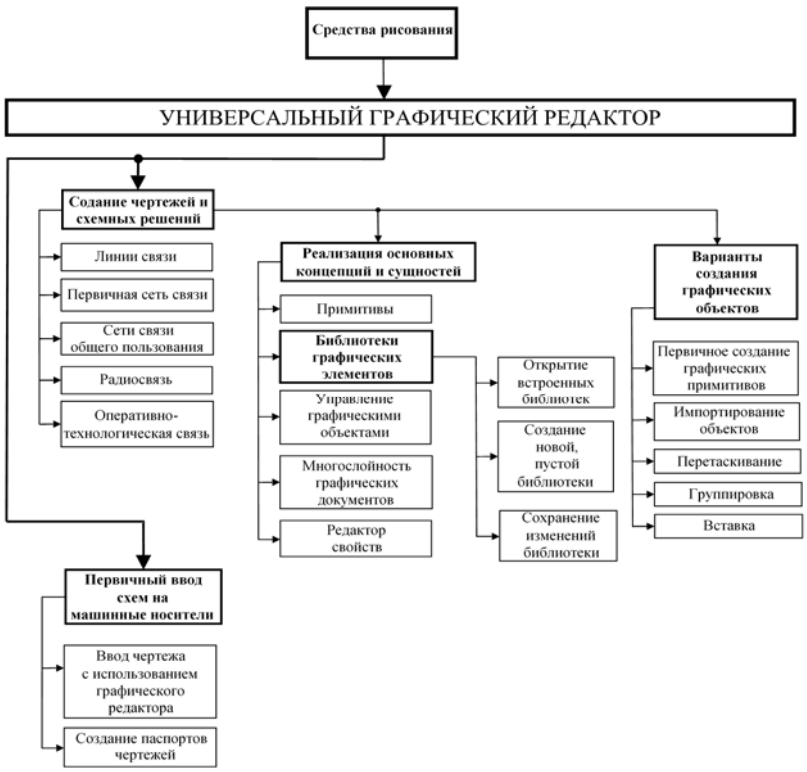


Рис. 4. Структурная схема модуля СР

Рассмотренная схема программной реализации позволяет решить все задачи, поставленные перед службой Ш по документообороту технической документации. Для работы с планами и схемами достаточно существующего графического редактора, входящего в модуль СР, который необходимо дополнить библиотекой специфических элементов, не представленных в базовой конфигурации. Модуль БЗ необходимо настроить согласно структуре подразделений и назначить права доступа пользователям (модуль ЗВ) по заданию руководства службы Ш, а также права доступа и порядок отображения информации смежным службам по решению Управления метрополитена.

Заключение

Использование современного комплексного подхода к организации электронного документооборота в Петербургском метрополитене позволит эффективно решать важные задачи, такие как:

– повышение эффективности ведения технической документации;

– ускорение проектирования технической документации;

– автоматизированный контроль правильности выполнения проектов на всех стадиях;

– проверка правильности работы систем методом моделирования;

– ускорение пусконаладочных работ (ввода систем в эксплуатацию);

– автоматизированный контроль правильности технической документации в процессе эксплуатации систем;

– увеличение эффективности автоматизированных систем управления (АСУ);

– улучшение информационного обеспечения руководителей всех уровней управления;

– повышение достоверности информационных потоков;

– ускорение процессов информационного обмена;

– применение электронной подписи;

– повышение эффективности управления.

Внедрение АРМ ВТД позволит экономить денежные ресурсы предприятия за счет сокращения времени движения в процессе обмена документами, удешевления процедуры подготовки, доставки, учета и хранения документов; гарантировать достоверность информации, повысить конфиденциальность и минимизировать риск финансовых потерь, отказаться от командировочных расходов.

Внедрение АРМ ВТД позволит экономить временные ресурсы за счет упрощения процедуры согласования и возможности удаленной работы с документами.

Немаловажным положительным аспектом внедрения АРМ ВТД является универсальность и оптимизация для нужд конкретной службы метрополитена, что в свою очередь позволяет без временных потерь на обучение и адаптацию технических специалистов замещать аналогичные должности в смежных службах метрополитена.

Литература

1. Развитие электронного документооборота в хозяйстве АТ / М.Н. Василенко, В.Г. Трохов, Д.В. Зуев, Д.В. Седых // Автоматика, связь, информатика. – 2015. – № 1. – С. 14–16.

2. Формализация алгоритмического описания систем обеспечения жизненного цикла железнодорожной автоматики и телемеханики / П.Е. Булавский, Д.С. Марков, В.Б. Соколов, Т.Ю. Константинова // Автоматика на транспорте. – 2015. – Т. 1, № 4. – С. 418–432.

3. Принципы организации электронного документооборота технической документации / М.Н. Василенко, Б.П. Денисов, П.Е. Булавский, Д.В. Седых // Транспорт Российской Федерации. – 2006. – № 7 (7). – С. 31–35.

4. Medina R., Meyers S., Fenner J. Document Workflow in the Enterprise // Doculabs, 2002.

5. Булавский П.Е., Марков Д.С. Методика оценки временных характеристик процессов электронного документооборота технической документации // Автоматика на транспорте. – 2016. – Т. 2, № 1. – С. 81–94.

6. BS EN 50129:2003. Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signaling.

7. Василенко М.Н., Кудрявцев В.В. Концепция построения единой автоматизированной системы электронного документооборота на устройства СЦБ // Автоматика, связь, информатика. – 2002. – № 9. – С. 2–5.

8. Проблемы внедрения отраслевого формата / Н.Н. Балуев, М.Н. Василенко, В.Г. Трохов // Автоматика, связь, информатика. – 2010. – № 3. – С. 2.

9. Седых Д.В. Интеграционные решения на основе отраслевого формата технической документации // Транспорт Урала. – 2016. – № 4 (51). – С. 52–57. – DOI: 10.20291/1815-9400-2016-4-52-57.

10. Wei Y.C., Linm C.H. A robust video text detection approach using SVM. Expert Syst. Appl. 10832–10840 (2012).

11. Ефанов Д. В. Некоторые аспекты развития систем функционального контроля устройств железнодорожной автоматики и телемеханики // Транспорт Урала. – 2015. – № 1. – С. 35–40.

12. Тележенко Т.А. Применение методов моделирования в системах автоматизированного проектирования // Известия Петербургского университета путей сообщения. – 2006. – № 2. – С. 66–72.

13. Тележенко Т.А. Автоматизированная система экспертизы схемных решений ЖАТ //Автоматика, связь, информатика. – 2009. – № 5. – С. 24–26.

14. Горбачев А.М. Автоматизация анализа, экспертизы и сверки технической документации системы железнодорожной автоматики и телемеханики // Известия Петербургского университета путей сообщения. – 2012. – № 4. – С. 73–78.

15. Kim K.I., Jung K., Kim J.H. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Trans. PAMI 25(12), 1631–1639 (2003).

16. Разработка обобщенной модели принципиальных электрических и монтажных схем для экспертизы и интеллектуальной корректировки / А.А. Матушев, Д.В. Седых, И.С. Ушаков // Материалы Юбилейной XV Санкт-Петербургской международной конференция «Региональная информатика – 2016», Санкт-Петербург, 26–28 октября 2016 г. – СПб. : СПОИСУ, 2016. – С. 305. – ISBN 978-5-906841-68-1.

17. Smith D., Field J., Learned-Miller E. Enforcing similarity constraints with integer programming for better scene text recognition, in Proceedings of CVPR (2011), pp. 73–80.

18. Матушев А.А. Программный комплекс для распознавания монтажной технической документации // Известия Петербургского университета путей сообщения. – 2015. – № 1. – С. 105–109.

19. Gllavata J., Ewerth R., Freisleben B. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients, in Proceedings of ICPR (2004).

20. Седых Д. В., Матушев А. А. Методы распознавания структуры монтажных схем железнодорожной автоматики и телемеханики // Автоматика на транспорте. – 2016. – Т. 2, № 4. – С. 552–563.

Prospects for the Introduction of an Automated Workplace for the Maintenance of Technical Documentation in the St. Petersburg Metro

D.V. Zuev, D.V. Sedyh, D.V. Somov
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
zuevdv@gmail.com, sedyhdmitriy@gmail.com, somov.sdv@yandex.ru

Abstract. **The issues of developing a set of tools for automated creation of technical documentation control of its correctness, testing the operability of the systems being designed and the organization of information exchange. As well as the possibility of creating modern, viable and competitive electronic document management systems that solve issues related to the needs of the transport complex on the distribution and processing of information flows.**

Keywords: **technical documentation, electronic document management, automation, schematic plant layout, two-line station plan, circuit diagrams, local area network.**

References

1. Development of electronic document management in the economy of the AT [Razvitiye elektronnogo dokumentooborota v khozyaystve AT] / M.N. Vasilenko, V.G. Trokhov, D.V. Zuev, D.V. Sedykh // Automation communication, informatics. – 2015. – № 1. – P. 14–16. (In Rus.).
2. Formalization of the algorithmic description of the systems for ensuring the life cycle of railway automation and telemechanics [Formalizatsiya algoritmicheskogo opisaniya sistem obespecheniya zhiznennogo tsikla zheleznodorozhnoy avtomatiki i telemekhaniki] / P.E. Bulavsky, D.S. Markov, V.B. Sokolov, T.Y. Konstantinova // Automation on Transport. – 2015. – Volume 1, № 4. – P. 418–432. (In Rus.).
3. Principles of organization of electronic document circulation of technical documentation [Printsipy organizatsii elektronnogo dokumentooborota tekhnicheskoy dokumentatsii] / M.N. Vasilenko, B.P. Denisov, P.E. Bulavsky, D.V. Sedykh // Transport of the Russian Federation. – 2006. № 7 (7). – Pp. 31–35. (In Rus.).
4. Medina R., Meyers S., Fenner J. Document Workflow in the Enterprise // Doculabs, 2002.
5. Bulavsky P.E., Markov D.S. A technique for estimating the temporal characteristics of processes of electronic document circulation of technical documentation [Metodika otsenki vremennykh kharakteristik protsessov elektronnogo dokumentooborota tekhnicheskoy dokumentatsii] // Automation on Transport. – 2016. – Volume 2, number 1. – P. 81–94. (In Rus.).
6. BS EN 50129:2003. Railway applications. Communication, signalling and processing systems. Safety related electronic systems for signaling.
7. Vasilenko M.N., Kudryavtsev V.V. The concept of constructing a single automated system of electronic document management for STS devices [Kontseptsiya postroye-niya yedinoy avtomatizirovannoy sistemy elektronnogo dokumentooborota na ustroystva STSB] // Automation, Communication and Informatics. – 2002. – No. 9. – Pages 2–5. (In Rus.).
8. Problems of introducing the industry format [Problemy vnedreniya otraslevogo formata] / N.N. Baluev, M.N. Vasilenko, V.G. Trokhov // Automation, communication, informatics. – 2010. – No. 3. – P. 2. (In Rus.).
9. Sedyh D.V. Integration solutions based on the industry format of technical documentation [Integratsionnyye resheniya na osnove otraslevogo formata tekhnicheskoy dokumentatsii] // Transport of the Urals. 2016. No. 4 (51). Pp. 52–57. – DOI: 10.20291 / 1815-9400-2016-4-52-57. (In Rus.).
10. Wei Y.C., Linm C.H. A robust video text detection approach using SVM. Expert Syst. Appl. 10832–10840 (2012).
11. Efanov D.V. Some aspects of development of systems of functional control of devices of railway automation and telemechanics // Transport of the Urals. – 2015 – No. 1. – P. 35–40. (In Rus.).
12. Telezhenko T.A. Application of modeling methods in CAD systems [Primeneniye metodov modelirovaniya v sistemakh avtomatizirovannogo proyektirovaniya] // Proceedings of Petersburg Transport University. – 2006. – № 2. – P. 66–72. (In Rus.).
13. Telezhenko T.A. Automated system of examination of circuit solutions of the automatic control system [Avtomatizirovannaya sistema ekspertizy skhemnykh resheniy ZHAT] // Automation, communication, informatics. – 2009. – № 5. – P. 24–26. (In Rus.).
14. Gorbachev A.M. Automation of analysis, examination and verification of technical documentation of the railway automation and telemechanics system [Avtomatizatsiya analiza, ekspertizy i sverki tekhnicheskoy dokumentatsii sistemy zheleznodorozhnoy avtomatiki i telemekhaniki] // Proceedings of Petersburg Transport University. – 2012. – № 4. – P. 73–78. (In Rus.).
15. Kim K.I., Jung K., Kim J.H. Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm. IEEE Trans. PAMI 25(12), 1631–1639 (2003).
16. Development of a generalized model of basic electrical and wiring diagrams for examination and intellectual correction [Razrabotka obobshchennoy modeli printsipial'nykh el-

ektricheskikh i montazhnykh skhem dlya ekspertizy i intellektual'noy korrektirovki] / A.A. Matushev, D.V. Sedyh, I.S. Ushakov // Materials of the Jubilee XV St. Petersburg International Conference "Regional Informatics – 2016", St.Petersburg, 26–28 October 2016, SPOISU, St. Petersburg, 2016. – P. 305. – ISBN 978-5-906841-68-1. (In Rus.).

17. Smith D., Field J., Learned-Miller E. Enforcing similarity constraints with integer programming for better scene text recognition, in Proceedings of CVPR (2011), pp. 73–80.

18. Matushev A.A. The software complex for recognition of the installation technical documentation [Programmnyy kompleks dlya raspoznavaniya montazhnoy tekhnicheskoy dokumentatsii] // Proceedings of Petersburg Transport University. – 2015. – No. 1. – P. 105–109. (In Rus.).

19. Gllavata J., Ewerth R., Freisleben B. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients, in Proceedings of ICPR (2004).

20. Sedyh D.V., Matushev A.A. Methods for recognizing the structure of wiring diagrams of railway automation and remote control [Metody raspoznavaniya struktury montazhnykh skhem zheleznodorozhnoy avtomatiki i telemekhaniki] // Automation on Transport. – 2016. – T. 2, № 4. – S. 552–563. (In Rus.).

Построение пользовательского интерфейса путем интеграции программ на языке R в систему Microsoft Visual Studio при обработке данных для задач геодезии

И.В. Коротеев, М.Я. Брынь
Петербургский государственный университет путей сообщения
Императора Александра I
Санкт-Петербург, Россия
KoroteevIlya94@gmail.com, 3046921@mail.ru

***Аннотация*. Система R является мощной платформой для ведения статистической обработки данных и их анализа, однако у нее есть свои минусы. Как правило, программы, написанные на языке R, не имеют удобного графического интерфейса, поэтому их применение и распространение для сторонних пользователей может быть затрудненно. Эта проблема может быть решена путем интеграции системы R в другую, более расположенную для создания интерфейсов систему. В статье рассматривается способ взаимодействия R-системы с платформой .Net Framework посредством интеграции в программу библиотеки R.Net. Этот подход позволяет создавать полномасштабные графические интерфейсы, одновременно используя все возможности языка R для статистического анализа и обработки данных. Приведен пример решения задач статистической обработки данных геодезической информации.**

***Ключевые слова:* R-система, R.Net, NuGet, Microsoft Visual Studio (MVS), CSV, статистический анализ, линейная регрессия, коэффициент корреляция, графический интерфейс пользователя.**

Введение

Система R включает в себя язык программирования сверхвысокого уровня для статической обработки данных и средства для работы с графикой. В настоящее время это один из мощнейших инструментов, предназначенных для обработки статических данных любой структуры.

Необходимо отметить, что R является объектно-ориентированным языком программирования. Кроме основного набора вычислительных пакетов, возможна установка дополнительных пакетов, которых насчитывается более 5000. Эти пакеты устанавливаются с официального сайта R [1].

Несмотря на все свои преимущества, R имеет достаточно ограниченный функционал в части предоставления пользователю графического интерфейса взаимодействия с системой.

При работе с R используются следующие способы:

- создание скриптов на языке R непосредственно из консоли разработчика, доступной после установки дистрибутива с официального сайта;
- работа в бесплатной среде разработки для языка R – RStudio;
- поиск и установка пакетов, позволяющих интегрировать язык R в другие системы.

В этой статье рассматривается третий способ работы с R.

В статье [2] рассматривались возможности интеграции другой математически ориентированной системы MATLAB с Microsoft Visual Studio. Интеграция происходила методом создания динамических библиотек (.dll) из исходного кода на MATLAB. Указанный подход позволял решить проблему построения графического интерфейса к программам MATLAB.

В нашей статье рассматривается возможность интеграции R с Microsoft Visual Studio на языке C# с помощью пакета R.Net. Продемонстрирована возможность создания эстетичного, интуитивного интерфейса для программы, созданной на основе системы R.

Microsoft Visual Studio является основной средой разработки приложений для операционной среды Windows, имеет обширный и разнообразный инструментарий для создания графического интерфейса пользователя. Кроме того, в MVS имеется удобный менеджер работы со сторонними пакетами. Этими достоинствами был продиктован выбор системы для интеграции с R.

Пакет R.Net является встроенным мостом для .Net Framework, предоставляющим доступ к статистическому языку R. Среда R.Net разработана Косей Эйбом и Перро Жан-Мишелем и работает в операционных системах Windows, Linux и MacOS.

Среда R.Net позволяет среде .Net Framework взаимодействовать с языком статистической обработки данных R в рамках единого процесса. Для работы со средой R.Net необходимы среда .Net Framework не ниже четвертой версии, а также установленный компилятор языка R.

Среду R.Net можно использовать со многими языками программирования, работающими с платформой .Net Framework со всеми установленными в ней языками программирования (C#, F#, Vb.net, IronPython) [3].

Настройка и установка среды R.Net в Visual Studio

Начиная с июля 2015 года разработчиками рекомендуется устанавливать среду R.Net с помощью NuGet – менеджера пакетов, предназначенного для работы с .NET платформой. Клиентские инструменты NuGet предоставляют возможность создавать и устанавливать различные пакеты (библиотеки) для работы с новым проектом.

Для этого необходимо установить NuGet с помощью следующих действий в VisualStudio: Инструменты -> Расширения и обновления (рис. 1).

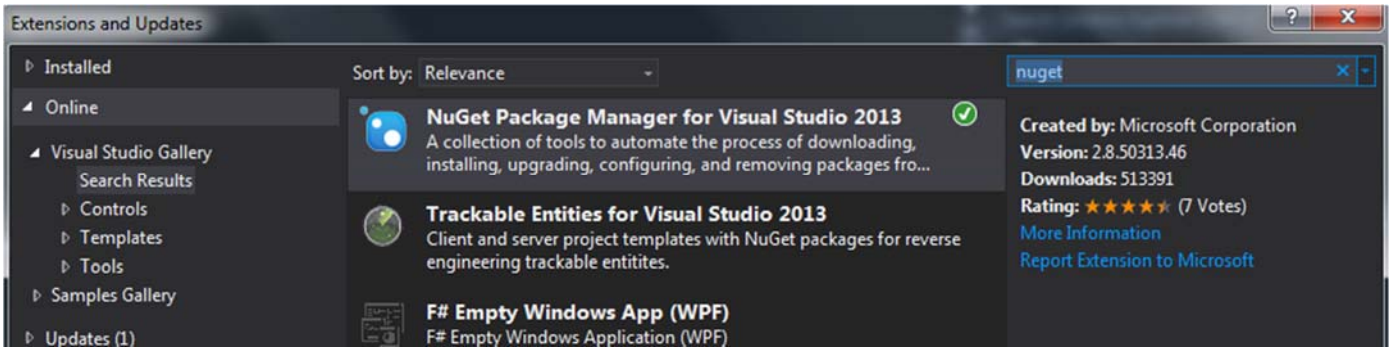


Рис. 1. Добавление менеджера пакетов NuGet

Среда R.Net может быть добавлена как зависимость для одного (рис. 2) или нескольких проектов в текущем проекте.

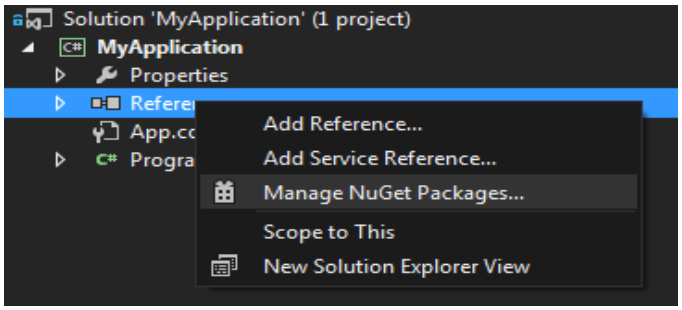


Рис. 2. Добавление R.Net в один проект

Следует отметить, что необходимо удалять ранее имевшиеся зависимости либо R.Net версии 1.5.5, либо более ранней версии. R.Net 1.5.13 использует другой идентификатор пакета: R.Net.Community. Необходимо убедиться, что используется последняя версия R.Net. Это можно сделать с помощью NuGet (рис. 3). NuGet добавит в проект несколько ссылок (RDotNetи RDotNet.NativeLibrary).

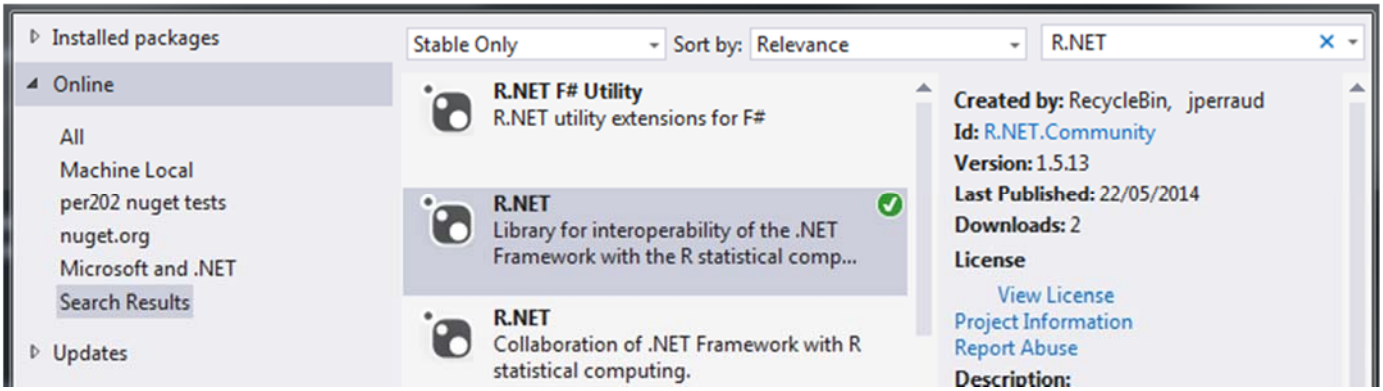


Рис. 3. Проверка версии R.Net

Использование возможностей R.Net

При разработке проекта следует придерживаться следующих правил. После установки необходимых переменных окружения извлекается один объект класса REngine, являющийся объектом-мостом к языку R в проекте. При обращении к этому объекту имеется возможность вызывать различные R-методы. Можно опустить вызов REngine.SetEnvironmentVatiable(), но рекомендуется явное объявление. SetEnvironmentVatiables просматривает реестр Windows в поисках установленного языка R.

```
staticvoidMain(string[] args)
{
REngine.SetEnvironmentVariables();
REngine engine = REngine.GetInstance();
{
```

Использование векторов

Обычно взаимодействие с REngine-объектом происходит посредством методов Evaluete, GetSymbol и SetSymbol.

- Evaluete() – позволяет однозначно передать синтаксис языка R обычной строкой в C# для вызова R функций, как если бы вы работали с R консолью;
- GetSymbol() – позволяет получить переменную типа языка R;
- SetSymbol() – позволяет установить переменную типа языка R.

Кроме того, REngine-объект имеет расширенные методы, такие как CreateNumericVector, CreateCharacterMatrix и другие для создания R-векторов или матриц. Ниже приводится пример использования базовых методов с числовыми векторами с помощью R.Net.

```
vare=engine.Evaluate("x <- 3");
NumericVectorx=engine.GetSymbol("x").AsNumeric();
engine.Evaluate("y <- 1:10");
NumericVectory=engine.GetSymbol("y").AsNumeric();
```

Использование R-фунций

R-функции и вызовы функций можно обозначить строками, эквивалентными их вызовам из языка R, с помощью метода Evaluate():

```
engine.Evaluate("cases <- expand.grid(x=c('a','b','c'),
y=1:3)")
vardf=engine.Evaluate("expand.grid(x=c('A','B','C'),
y=1:3)").AsDataFrame()
```

Следует отметить, что вызов функции, создание строки и вызов метода Evaluate может оказаться неудобным в случаях, когда используется большое количество входных данных. Для упрощения имеется возможность прямого (рефлексивного) вызова функции.

```
varmyFunc=engine.Evaluate("function(x, y) {
expand.grid(x=x, y=y) }").AsFunction();
```

```
varv1=engine.CreateIntegerVector(new[]{1,2,3});
```

```
varv2=engine.CreateCharacterVector(new[]{"a","b","c"});
vardf=myFunc.Invoke(newSymbolicExpression[]{v1,v2})
.AsDataFrame();
```

R.Net1.5.10 включает в себя множество улучшений по поддержке вызовов функций прямо из C#, что позволяет уменьшить количество вызовов REngine.Evaluate, а также упростить манипуляции со строками:

```
varexpand-
Grid=engine.Evaluate("expand.grid").AsFunction();
```

```
vard=newDictionary<string,SymbolicExpression>();
d["x"]=v1;
d["y"]=v2;
df=expandGrid.Invoke(d).AsDataFrame();
```

Пример взаимодействия R.Net и Microsoft Visual Studio

В качестве примера взаимодействия R и MVS рассматриваются следующие задачи:

- построение линейной регрессии зависимости нормальной и геодезической высоты точек плоскости с помощью функционала системы R;
- добавление в программу графического пользовательского интерфейса, построенного с помощью возможностей платформы .Net и MVS.

Для демонстрации простоты разработки подобной программы будет создано простейшее приложение Windows Forms с минимальным набором графических элементов. На рис. 4 приведен полный графический интерфейс пользователя разработанной программы:

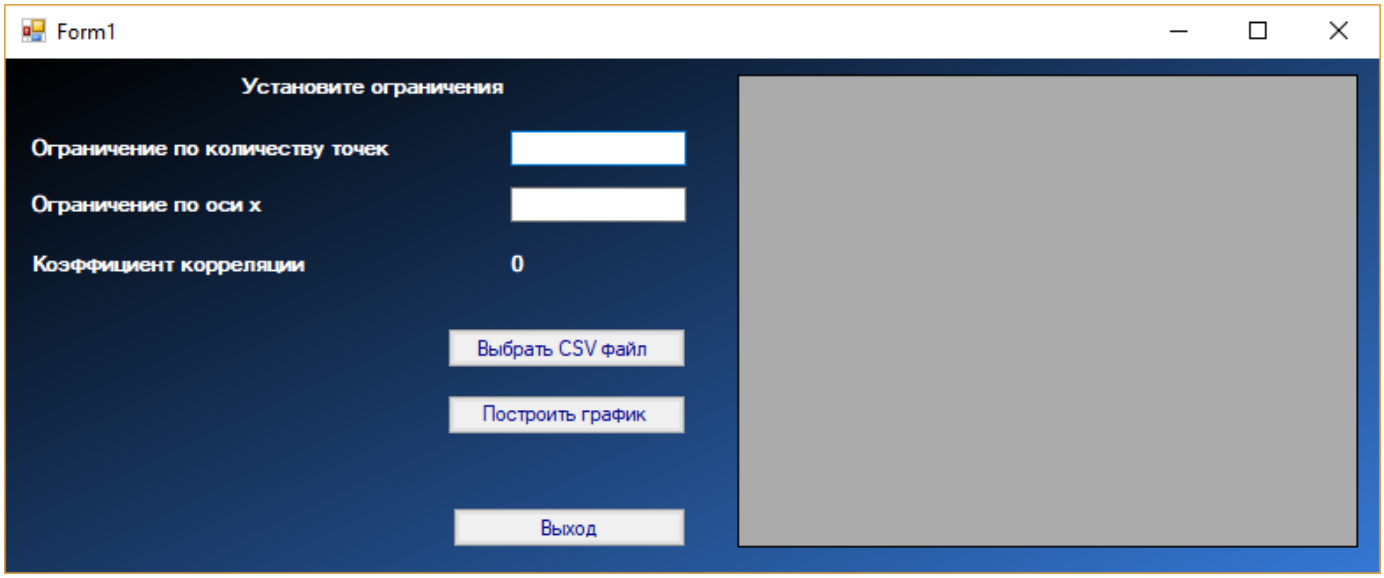


Рис. 4. Графический интерфейс пользователя

Интерфейс состоит из двух полей ввода данных:

• количество точек, рассматриваемых в регрессионной модели (количество точек на графике);

• максимальный размер данных (ограничение графика по оси x).

После ввода данных производится проверка на их корректность: если пользователь ввел не числа, появляется окно с предупреждением. После построения линейной регрессии пользователь увидит коэффициент корреляции рассматриваемых величин без оценки статистической значимости.

Кнопка «Выбрать CSV-файл» вызывает файловый менеджер и позволяет загрузить подготовленный файл с данными в формате CSV. После выбора таблица отображается в форме DataGrid, и пользователь может наблюдать загруженные данные в удобном формате. Функция в цикле заполняет объект данных R DataFrame и возвращает его в основной класс приложения. В случае отправки неверно заполненного файла с данными приложение выдаст пользователю предупреждение [4].

Кнопка «Построить график» запускает основную функцию программы, в которой происходит все взаимодействие с системой. Ниже приводится исходный текст программы на языке C#.

```
public static double getRPlane(intsizeCount, intmaxSize,
DataFrame dataset) {

double pearsonCoefficient = 0.00f;

bool isCreate = true;

REngine.SetEnvironmentVariables();

REngine engine = REngine.GetInstance();

List<double> Size = newList<double>();

List<double> Population = newList<double>();

for (inti = 0; i<dataset.RowCount; ++i){

if (i<sizeCount || sizeCount == 0)

{

try  {

double currentSize = double.Parse(dataset[i, da-
taset.ColumnCount-2].ToString());

double currentPopulation = double.Parse(dataset[i, da-
taset.ColumnCount - 1].ToString());

if (currentSize<= maxSize || maxSize == 0) {

Size.Add(currentSize);

Population.Add(currentPopulation);

}

}

catch  {  isCreate = false;

MessageBox.Show(@"Неверныйформатячейки");

break;          }

        }

   }

if (isCreate) {

NumericVector sizeVector = en-
gine.CreateNumericVector(Size);

engine.SetSymbol("yVector", sizeVector);
```

```
    NumericVector populationVector = engine.CreateNumericVector(Population);
    engine.SetSymbol("xVector", populationVector);
    engine.Evaluate("reg<- lm(yVector~xVector)");
    engine.Evaluate("plot(yVector~xVector)");
    engine.Evaluate("abline(reg)");
    var corellation = engine.Evaluate("cor(xVector, yVector, method = c('pearson'))");
    var corellationSize = corellation.AsNumeric();
    pearsonCoefficient = corellationSize.ElementAt(0); }
    return pearsonCoefficient;
}
```

Как видно, в параметры функции передаются количество точек, максимальное значение параметра x, а также созданный функцией openCSV объект DataFrame.

Происходит инициализация переменных окружающей среды и создание объекта взаимодействия с R – REngine. Затем посылается запрос на выбор CSV-файла с исследуемыми массивами данных. В зависимости от введенных пользователем данных ограничивается максимальное количество точек или отбрасываются точки с превышающими определенный порог значениями. В случае игнорирования ограничений будет исследован весь массив данных.

Два C#-объекта List, а именно размер и количество, в цикле заполняются данными из объекта DataFrame CSV-файла. Далее создаются и устанавливаются числовые R-векторы xVector и yVector. Спомощью R-функций lm() строится линейная регрессионная зависимость, а с помощью функций plot() и abline() происходит построение графика.

В конце с помощью R-функции cor() рассчитывается коэффициент корреляции, и после преобразований с типами данных значение коэффициента возвращается в пользовательский интерфейс [5].

В примере рассматриваются геодезические и нормальные высоты некоторой географической плоскости. Различные методы решения подобных задач другими средствами рассмотрены в [6–8].

Для исследования пользователь может добавить произвольно подготовленный CSV-файл, удовлетворяющий следующим условиям:

- последние два столбца CSV-файла содержат данные для исследования;
- первая строка файла является названиями колонок.

Формат CSV-файла определен в [9].

Можно воспользоваться программой Microsoft Excel и сохранить таблицу в формате CSV.

Фрагмент исходного файла данных высот точек плоскости представлен в таблице.

ТАБЛИЦА. Содержание исходного файла данных

PTS	H geodesic	h normal
S001	10,915	2,625
S002	25,156	16,785
S003	10,598	2,368
S004	18,273	9,990
S005	37,727	29,309
S006	23,281	15,052
…	…	…
S019	18,723	10,892
S020	11,523	3,969
S021	44,273	36,454

На рисунке 5 показаны введенные ограничения и данные для построения регрессий.

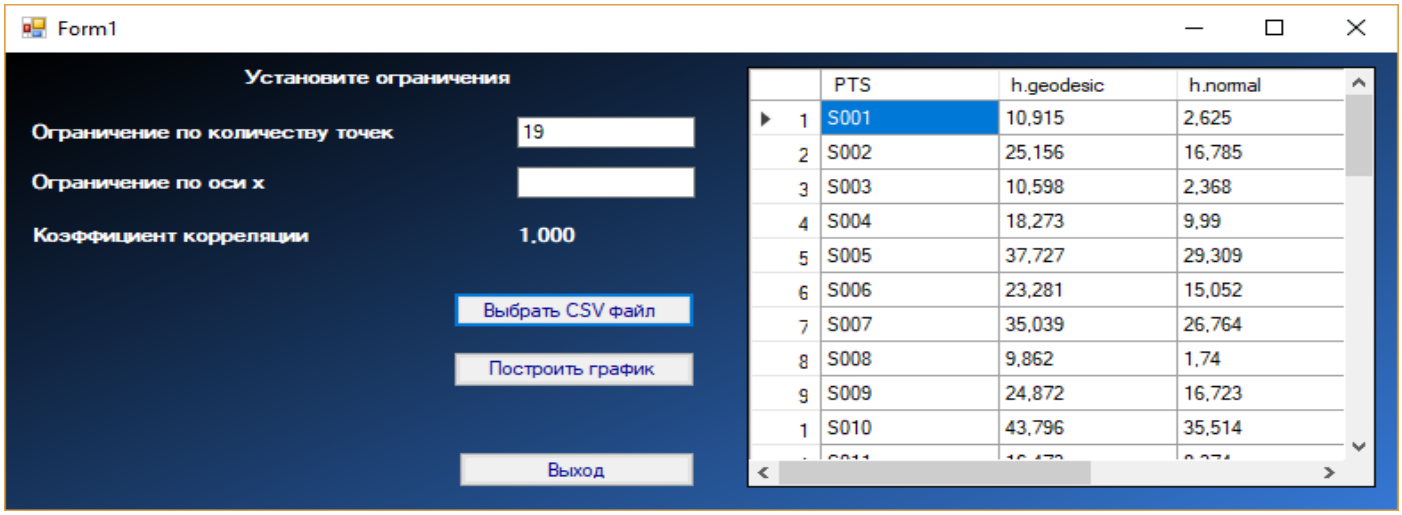


Рис. 5. Исходные данные точек

После запуска построения графика в строке «Коэффициент корреляции» можно увидеть расчет коэффициента без статистической значимости. В результате выполнения программы получается график соответствующей линейной регрессии (рис. 6).

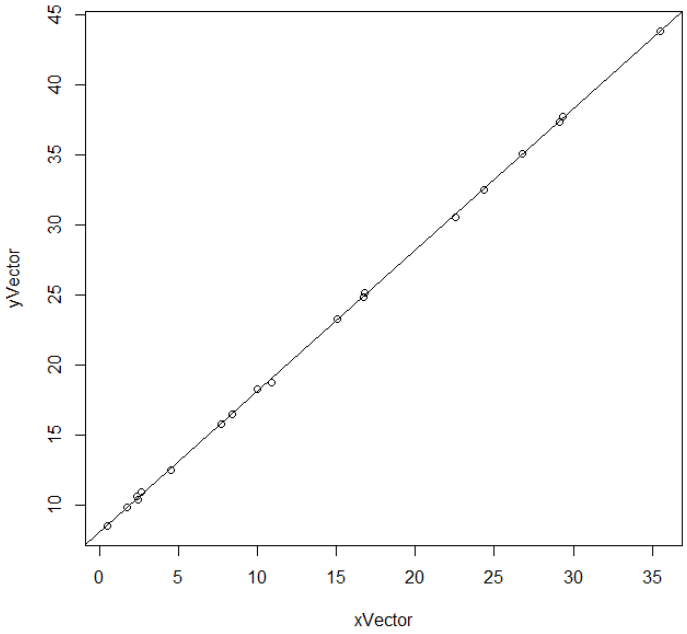


Рис. 6. Линейная регрессия высот точек плоскости

Как видим, мы получили графический интерфейс пользователя для решения задач статистической обработки данных геодезической информации.

Дополнительную информацию по технологии работы в рассматриваемых инструментальных средствах можно найти в работах [10–12].

Заключение

1. R.Net позволяет использовать R-переменные, вызывать R-функции, использовать все возможности статического анализа R в программе, написанной на C#. Продемонстрировано применение предлагаемой методики для решения задач статистической обработки данных геодезической информации.

2. Метод трансляции библиотекой-мостом аналогичен подобному методу в системе MATLAB, рассмотренному в [2], однако R не имеет возможности создавать независимое консольное приложение. Использование этого метода, в отличие от других методов, указанных в данной статье, позволяет создавать полномасштабные графические интерфейсы и Windows-приложения.

Литература

1. Красновидов А.В. Инструментальные средства информационных систем : учеб. пособие / А.В. Красновидов, С.Г. Свистунов, П.А. Новиков. – СПб. : ФГБОУ ВПО ПГУПС, 2015. – 48 с.

2. Методы интеграции инструментальных систем в процессе разработки безопасных приложений / С.Е. Ададуров, А.В. Красновидов, А.Д. Хомоненко, И.В. Коротеев // Проблемы информационной безопасности. Компьютерные системы. – 2017. – № 4. – С. 80–86.

3. Документация R.NET [Электронный ресурс]. – Режим доступа: http://www.jmp75.github.io/rdotnet/ дата посещения 2.03.2018.

4. Оливер Функе. Use R in C#: Create a data plot and save it as png [Электронный ресурс]. – Режим доступа: https://coders-corner.net/2015/11/22/use-r-in-c-create-a-data-plot-and-save-it-as-png/ дата посещения 29.04.2018.

5. Мастицкий С. Анализ и визуализация данных. Классические методы статистик: коэффициент корреляции [Электронный ресурс]. – Режим доступа: https://r-analytics.blogspot.ru/2012/09/blog-post_6280.html#. WvFnPYiFOUl / дата посещения 28.04.2018.

6. Программный комплекс для мониторинга деформаций особо опасных объектов / М.Я. Брынь, А.Д. Хомоненко, В.П. Бубнов и др. // Проблемы информационной безопасности. Компьютерные системы. – 2014. – № 1. – С. 36–41.

7. Программный комплекс автоматизированного геодезического мониторинга искусственных сооружений для высокоскоростной железнодорожной магистрали «Москва – Казань – Екатеринбург» / В.П. Бубнов, А.А. Никитчин, С.А. Сергеев // Интеллектуальные технологии на транспорте. – 2015. – № 4. – С. 27–34.

8. Технология удаленного мониторинга пространственного положения пилотируемого летательного аппарата и состояние его бортовых систем в режиме реального времени / С.В. Кулешов, А.А. Зайцева, А.Ю. Аксенов // Интеллектуальные технологии на транспорте. – 2016. – № 2(6). – С. 43–49.

9. ГОСТ Р 51794–2001. Системы координат. Методы преобразований координат определяемых точек.– М. : Госстандарт России, 2001. – 11 с.

10. Инг Бей, Джифенг Ксу. Взаимодействие MATLAB с ANSI C, Visual C++, Visual Basic и Java. – М. : Вильямс, 2005. – 207 с.

11. Hejlsberg A. DEV223: The .NET Language Integrated Query (LINQ) Overview // Microsoft Tech Ed Developers, 7–10 Nov. 2006. Barcelona, Spain. – Режим доступа: http://msmvps.com/blogs/vandooren/archive/2006/11/07/Tech_2D00_Ed-developers-Barcelona_3A00_-Tuesday.aspx.

12. R: A Language and Environment for Statistical Computing Reference Index. The R Core Team. Version 3.5.0 (2018-04-23). – Режим доступа: http://lib.stat.cmu.edu/ R/CRAN/doc/manuals/fullrefman.pdf.

Creation of the User Interface by Integration of Programs at Language R with the Microsoft Visual Studio System at Data Processing for Problems of Geodesy

I.V. Koroteev, M.Ya. Bryn
Emperor Alexander I St.-Petersburg, Russia
KoroteevIlya94@gmail.com, 3046921@mail.ru

Abstract: **The system R is the powerful platform for conducting statistical data processing and their analysis, however she has minuses. Often, the programs written in language R have no convenient graphic interface, both their application, and distribution for third-party users can be complicated. This problem can be solved by integration of system R into another, more located for creation of interfaces, system. In article the way of interaction R-system with the .Net Framework platform, by means of integration into the program of R.Net library is considered. This approach allows create full-scale graphic interfaces, at the same time using all opportunities of language R for the statistical analysis and data processing. The example of use of approach for the solution of problems of statistical data processing of geodetic information is given.**

Keyword: **R-system, R.Net, NuGet, Microsoft Visual Studio (MVS), CSV, statistical analysis, linear regression, correlation coefficient, graphical user interface.**

REFERENCES

1. Krasnovidov A.V. Tools of information systems: study guide [instrumentalnye sredstva informacionnyh sistem: uchebnoe posobie], SBP, FGBOU VPO PSTU, 2015, 48 p.

2. Adadurov S.Ye., Krasnovidov A.V., Khomonenko A.D., Koroteev I.V. Methods of integration of instrumental systems in the process of developing secure applications [Metody integracii instrumentalnyh system v processe razrabotki bezopasnyh prilozhenij], Problems of information security. Computer systems. [Problemy informacionnoj bezopasnosti kompyuternye sistemy], 2017, no. 4, pp. 80–86.

3. Documentation R.NET [Dokumentaciya r net]. Available at: http://www.jmp75.github.io/rdotnet/ (accessed 2 March 2018).

4. Use R in C #: Create a data plot and save it as png [Ispolzovanie R v c#: sozdanie grafika dannyh i sohranenieego-v-formate-png sohranenie ego v formate png]. Available at: https://coders-corner.net/2015/11/22/use-r-in-c-create-a-data-plot-and-save-it-as-png/ (accessed 2 April 2018).

5. Analysis and visualization of data Classical statistical methods: Coefficient of correlation [analiz i vizualizaciya dannyh klassicheskie metody statistik koehfficient korrelyacii]. Available at: https://r-analytics.blogspot.ru/2012/09/blog-post_6280.html#.WvFnPYiFOUl (accessed 28 April 2018).

6. Bryn M.Ya., Khomonenko A.D., Bubnov V.P., Nikitchin A.A., Sergeev S.A., Novikov P.A., Titov A.I. Software for monitoring strain especially dangerous objects [Programmnyi kompleks dlia monitoringa deformatsii osobo opasnykh obieektov] // Information Security Problems. Computer Systems *[Problemy informatsionnoi bezopasnosti Kompiuternye sistemy]*. 2014. No 1. – Pp. 36–41.

7. Bubnov VP, Nikitin AA, Sergeev SA Software complex of automated geodetic monitoring of artificial structures for the high-speed railway line "Moscow–Kazan–Yekaterinburg" [Programmnyj kompleks avtomatizirovannogo geodezicheskogo monitoringa iskusstvennyh sooruzhenij dlya vysokoskorostnoj zheleznodorozhnoj magistrali moskva kazan ekaterinburg], Intelligent technologies in transport [Intellektualnye tekhnologii na transporte], 2015, no. 4, pp. 27–34.

8. The technology of remote monitoring of the spatial position of a manned aircraft and the state of its on-board systems in real time [Tekhnologiya udalennogo monitoringa prostranstvennogo polozheniya pilotiruemogo letatelnogo apparata i sostoyaniya ego bortovyh sistem v rezhime realnogo vremeni] / S.V. Kuleshov, A.A. Zaytseva, A.Yu. Aksenov // Intelligent technologies in transport [Intellektualnye tekhnologii na transporte], 2016, no. 2 (6), pp. 43–49.

9. GOST R 51794–2001: Coordinate systems. Methods for transforming the coordinates of determined points [Gost r 51794 2001 sistemy koordinat metody preobrazovanij koordinat opredelyaemyh tochek]. Moscow, Gosstandart of Russia, 2001, 11 p.

10. Ing Bay, Gifeng Xu. Interaction of MATLAB with ANSI C, Visual C ++, Visual Basic and Java, M. : Williams, 2005, 207 pp.

11. Hejlsberg A. DEV223: The .NET Language Integrated Query (LINQ) Overview // Microsoft Tech Ed Developers, 7–10 Nov. 2006. Barcelona, Spain. http://msmvps.com/blogs/vandooren/archive/2006/11/07/Tech_2D00_Ed-developers-Barcelona_3A00_-Tuesday.aspx.

12. R: A Language and Environment for Statistical Computing Reference Index. The R Core Team. Version 3.5.0 (2018-04-23). – http://lib.stat.cmu.edu/ R/CRAN/doc/manuals/fullrefman.pdf.

Распознавание зашумленных текстовых символов с помощью обучаемой нейронной сети

А.В. Красновидов, А.С. Алексеев
Петербургский государственный университет путей сообщения Императора Александра I
Санкт-Петербург, Россия
alexkrasnovidow@mail.ru, affectum@hotmail.com

***Аннотация*. Системы компьютерного зрения и распознавания образов применяются при решении широкого спектра задач управления, оповещения, в военных и медицинских целях. Такие системы выполняют преобразование исходных многомерных данных (цветных растровых изображений) в набор обобщенных выходных данных (классов объектов, гипотез, функций принадлежности). На начальных этапах такого преобразования часто требуется найти графические примитивы – окружности, эллипсы, прямоугольники, кривые, прямые и пр. – на сложном зашумленном изображении и построить карту графических примитивов.**

В статье рассматриваются различные способы выделения таких примитивов, анализируются их достоинства и недостатки. Показаны преимущества использования обучаемых нейронных сетей для решения задач выделения графических примитивов. С помощью системы Matlab построена модель нейронной сети для распознавания 26 символов латинского алфавита. Проверена работоспособность построенной модели.

***Ключевые слова:* нейронные сети, модель графических примитивов, Matlab, NeuralNetworkToolbox.**

Введение

В повседневной практике часто возникает задача распознавания различных изображений, таких как фотографии – для идентификации личности, номеров автомобилей или вагонов и т. п. В силу различных причин эти изображения могут подвергаться воздействию мешающих факторов (помех), что ведет к их искажениям. Например, номера вагонов могут иметь следующие искажения: затертые сверху (рис. 1, а); вторая половина букв залита краской (рис. 1, б) неровно нанесены [1].

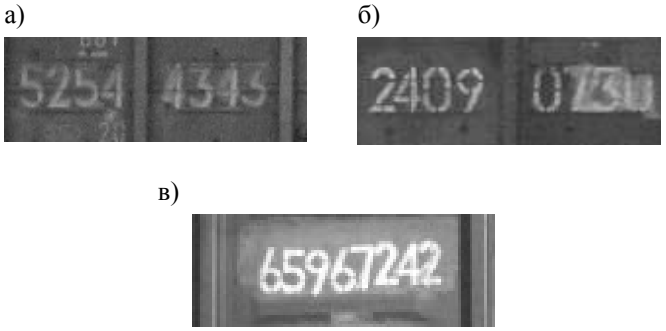


Рис. 1. Искаженные изображения номеров вагонов

В основу методологии распознавания изображений положен следующий принцип: изображение состоит из набора взаимосвязанных элементов. Каждый из элементов в свою очередь состоит из набора взаимосвязанных подэлементов, те также делятся на подэлементы нового уровня и так далее. На самом нижнем уровне изображение состоит из набора графических примитивов – простейших элементов, неделимых и легко распознаваемых. В качестве примитивов, как правило, выступают линии, отрезки дуг, кривые различной формы. В зависимости от конкретной задачи для упрощения и ускорения распознавания применяются также более сложные примитивы, например углы, образуемые пересечением линий, Т-образные конструкции, круги, стрелки и другие. Таким образом, процесс распознавания изображений можно разделить на три основных этапа:

- предварительная обработка изображения;
- векторизация и выделение примитивов;
- анализ изображения.

Отсюда следует, что в основе задачи распознавания изображений лежит выделение примитивов.

Анализ способов выделения графических примитивов

В настоящее время существует ряд продуктов, решающих задачу выделения графических примитивов. Для решения этой задачи они используют различные алгоритмы. Наиболее распространенными в настоящее время являются:

- преобразование Хафа;
- преобразование Радона;
- нейронная сеть.

Преобразование Хафа позволяет находить на монохромном изображении плоские кривые, заданные параметрически, например: прямые, окружности, эллипсы и т. д. Монохромным изображением считается изображение, состоящее из точек двух типов: фоновых точек и точек интереса. Задача преобразования Хафа состоит в выделении кривых, образованных точками интереса [2].

Преобразование Радона используется в компьютерной рентгеновской томографии и радиоастрономии для восстановления изображений по некоторому набору их проекций, являющихся прямыми линиями. Преобразование

Родона обладает свойством подавления шумов, поскольку вдоль этих прямых производится интегрирование. За счет этого отношение сигнал/шум в пространстве параметров выше, чем на исходном изображении, и обнаружение прямых обладает большей достоверностью [3].

Нейронная сеть является обучаемой математической моделью. Тип и структура такой сети выбирается и синтезируется для решения определенного типа задач. Обучение нейронной сети производится на основе информации о решении задачи экспертом или данных о решении задач в прошлом. Таким образом, для обучения сети нужно подобрать представительные данные и запустить алгоритм обучения, автоматически воспринимающий структуру введенных данных [4]. Результаты анализа разных способов выделения графических примитивов приведены в таблице 1.

ТАБЛИЦА 1. Способы выделения графических примитивов

Подход	Преимущества	Недостатки
Преобразование Хафа	Полное покрытие возможных состояний и положения объекта (за счет полного перебора в стандартном алгоритме). Модифицируемость алгоритма, что позволяет сократить время полного перебора без потери существенной информации	Предназначен для поиска прямых и окружностей
Преобразование Радона	Инвариантность по отношению к качеству изображения. Применяемый математический аппарат позволяет легко переходить к другим видам преобразований (аффинным, Фурье)	Предназначен для поиска прямых и окружностей. Сложность реализации
Нейронная сеть	Возможность выделения графических примитивов сложной структуры	Необходимость большого числа обучений сети и наличия представительной выборки данных. Ориентация на выделение заранее определенного класса изображений

Анализ таблицы 1 показывает, что нейронные сети могут распознавать примитивы более сложной структуры по сравнению с примитивами, распознаваемыми с помощью преобразований Хафа и Радона. Это может значительно упростить последующий анализ всего изображения в целом. С другой стороны, преобразования Хафа и Радона обладают большей универсальностью, так как выделяют элементарные (атомарные) примитивы, на которые может быть разложено изображение любой сложности.

На практике системы распознавания образов, как правило, ориентированы на распознавание какого-то определенного класса изображений. В этом случае использование нейронных сетей представляется более предпочтительным [5]. Примером такой задачи может служить задача распознавания текстов, состоящих из отдельных символов. В нашей статье рассматривается нейронная сеть распознавания 26 символов латинского алфавита.

НЕЙРОННАЯ СЕТЬ ДЛЯ РАСПОЗНАВАНИЯ СИМВОЛОВ

Контролируемые нейронные сети, или обучение с учителем, поддерживают три типа сетей: прямого распространения, радиально-базисные и динамические. Неконтролируемые сети, или обучающиеся без учителя, включают в себя такие сети, как конкурентные слои и самоорганизующиеся карты. Для распознавания символов подходят сети прямого распространения, потому что они включают в себя двухслойные сети с прямой связью, у которых диапазон выходных сигналов определен от 0 до 1 [6; 7].

Neural Network Tolbox (NNT) включает в себя функции командной строки и приложения для создания, обучения и моделирования нейронных сетей [8]. NNT поддерживает различные архитектуры контролируемых и неконтролируемых нейронных сетей. В настоящей статье исследование нейронной сети выполнено с помощью системы MATLAB (Matrix Laboratory), которая применяется для решения задач математических вычислений, моделирования сложных систем. Она имеет в своем составе одноимённый язык программирования [9].

В анализе использовался графический интерфейс NeuralNetworkToolbox, который позволяет, не обращаясь к командному окну системы MATLAB, выполнять создание, обучение, моделирование, а также импорт и экспорт нейронных сетей и данных, используя только инструментальные возможности GUI-интерфейса. Вызов GUI-интерфейса NNTool возможен либо командой nntool из командной строки, либо из окна запуска приложений Launch Pad с помощью опции NNTool из раздела Neural Network Toolbox. После вызова на экране терминала появляется окно **Network/Data Manager**, где есть возможность управлять сетью и данными (рис. 2) [10].

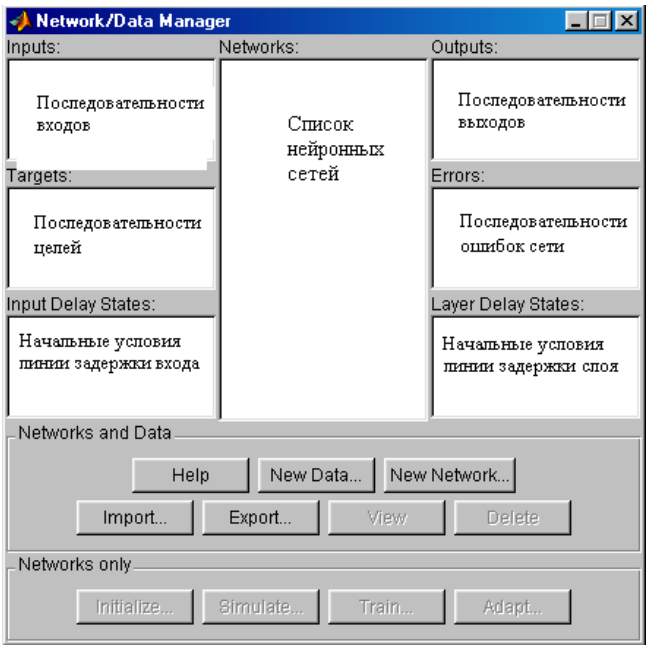
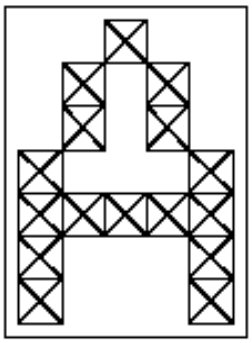


Рис. 2. GUI-интерфейс NNTool

Программа, разработанная в среде Matlab, иллюстрирует, как распознавание символов может быть реализовано в сети с обратным распространением. В качестве примера рассматривалась нейронная сеть для распознавания 26 символов латинского алфавита.

В качестве датчика используется система распознавания, которая выполняет оцифровку каждого символа в поле зрения. В результате каждый символ представлен шаблоном размера 5×7. Например, символ А может быть представлен, как это показано на рис. 3, *а* и *б*.

а)



б)

0	0	1	0	0
0	1	0	1	0
0	1	0	1	0
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

Рис. 3. Представление символа А

На рис. 4 показано искаженное по перечисленным выше причинам представление символа А.

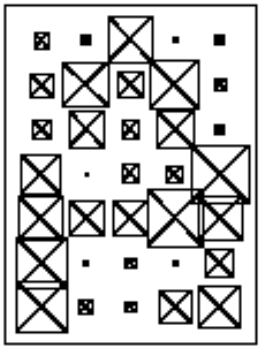


Рис. 4. Искаженный символ А

Проектируемая нейронная сеть должна точно распознавать идеальные векторы входа и с максимальной точностью воспроизводить зашумленные векторы. На вход сети поступает вектор входа с 35 элементами; вектор выхода содержит 26 элементов, только один из которых равен 1, а остальные – 0. Правильно функционирующая сеть должна ответить вектором со значением 1 для элемента, соответствующего номеру символа в алфавите.

Кроме того, сеть должна быть способной распознавать символы в условиях действия шума. Предполагается, что шум – это случайная величина со средним значением 0 и стандартным отклонением, меньшим или равным 0,2. Для работы нейронной сети требуется 35 входов и 26 нейронов в выходном слое. Для решения задачи использовалась двухслойная нейронная сеть с логарифмическими сигмоидальными функциями активации в каждом слое. Такая функция активации выбрана потому, что диапазон выходных сигналов для этой функции определен от 0 до 1, и этого достаточно, чтобы сформировать значения выходного вектора. Структурная схема такой нейронной сети показана на рис. 5 [11].

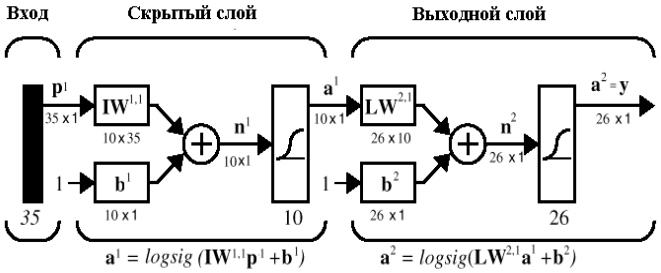


Рис. 5. Структурная схема нейронной сети

Скрытый слой имеет 10 нейронов. Если при обучении сети возникнут затруднения, то можно увеличить количество нейронов этого уровня. Сеть обучается так, чтобы сформировать единицу в единственном элементе вектора выхода, позиция которого соответствует номеру символа, и заполнить остальную часть вектора нулями.

Наличие шумов может приводить к тому, что сеть не будет формировать вектора выхода, состоящего точно из единиц и нулей. Поэтому по завершении этапа обучения выходной сигнал обрабатывается М-функцией compet, которая присваивает значение 1 единственному элементу вектора выхода, а всем остальным – значение 0. В результате формируется массив векторов входа для алфавита размером 35×26 с шаблонами символов алфавита и массив целых векторов. Двухслойная нейронная сеть создается с помощью команды newff, как это показано ниже [12]:

```
S1 = 10;
net = newff(minmax(alphabet),[S1 S2],{'logsig' 'logsig'},'traingdx');
net.LW{2,1} = net.LW{2,1}*0.01;
net.b{2} = net.b{2}*0.01;
```

Полученная в результате выполнения команды нейронная сеть имеет структуру, показанную на рисунке 6.

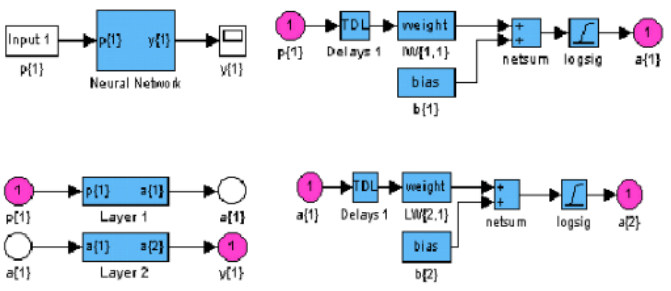


Рис. 6. Структура нейронной сети

Чтобы создать нейронную сеть, которая может обрабатывать зашумленные векторы входа, выполнено обучение сети на идеальных и на зашумленных векторах. Сеть первоначально обучается в отсутствие шума с максимальным числом циклов обучения 5000 либо до достижения допустимой средней квадратичной погрешности, равной 0,1 (рис. 7).

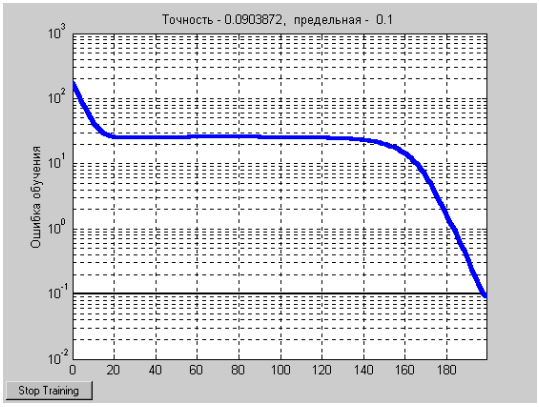


Рис. 7. Обучение сети в отсутствие шума

Для проектирования нейронной сети, не чувствительной к воздействию шума, необходимо обучить сеть с применением двух идеальных и двух зашумленных копий векторов алфавита. Целевые векторы состоят из четырех копий векторов. Зашумленные векторы имеют шум со средним значением 0,1 и 0,2. Это обучает нейрон правильно распознавать зашумленные символы и в то же время хорошо распознавать идеальные векторы. При обучении с шумом максимальное число циклов обучения необходимо сократить до 300, а допустимую погрешность увеличить до 0,6 (рис. 8).

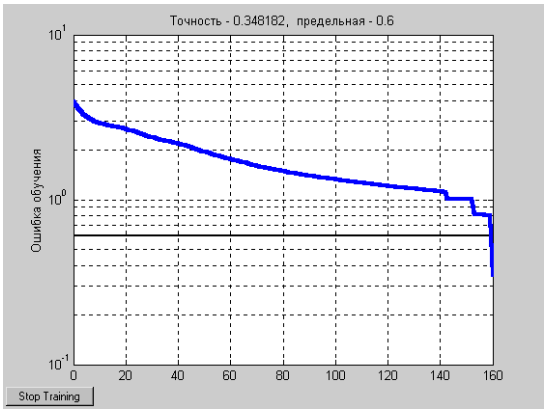


Рис. 8. Обучение сети при наличии шума

Поскольку нейронная сеть обучалась при наличии шума, то имеет смысл повторить ее обучение без шума для гарантии того, что идеальные векторы входа классифицируются правильно и избежать погрешности при отображение искомой буквы. Если необходима более высокая точность распознавания, сеть может быть обучена либо в течение более длительного времени, либо с использованием большего количества нейронов в скрытом слое. Можно также увеличить размер векторов, чтобы пользоваться шаблоном с более мелкой сеткой, например 10×14 точек вместо 5×7.

В качестве примера работы нейронной сети рассмотрено распознавание символа I. Для этого формируется зашумленный вектор входа для символа I (рис. 9). Ниже показан пример формирования зашумленного вектора входа:

```
noisyJ = alphabet(:,9) + randn(35,1)*0.2;
plotchar(noisyI);
```

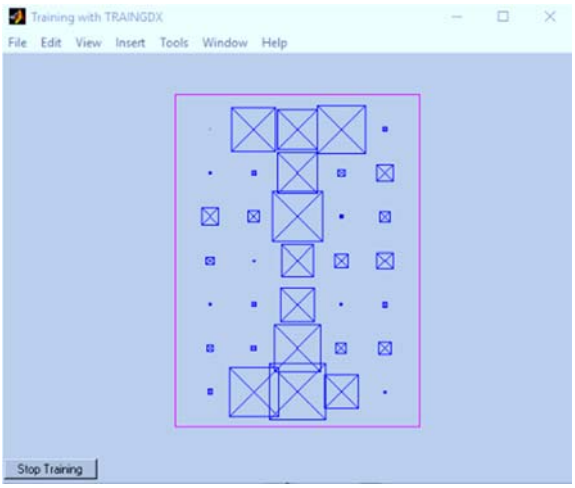


Рис. 9. Зашумленный символ I

Далее символ I восстанавливается по данному зашумленному образу (рис. 10). Фрагмент программы, восстанавливающий символ I, показан ниже:

```
A2 = sim(net,noisyJ);
A2 = compet(A2);
answer = find(compet(A2) == 1)
answer =    10
plotchar(alphabet(:,answer));
```

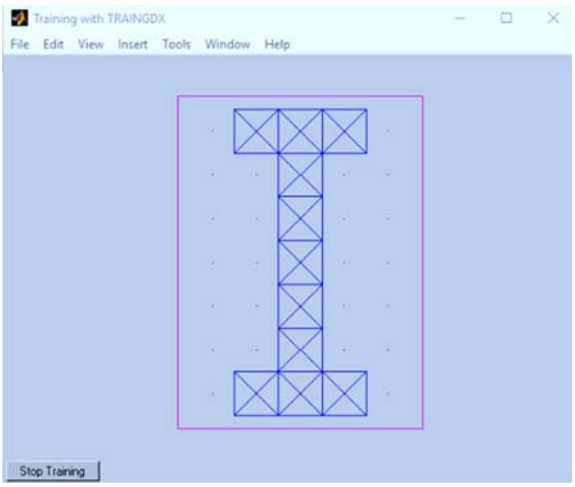


Рис. 10. Восстановленный символ I

Таким образом, нейронная сеть выделила 9 правильных элементов и восстановила символ I без ошибок, что подтверждает работоспособность предлагаемого метода.

Заключение

1. Нейронная сеть может быть использована для выделения графических примитивов, имеющих более сложную структуру, чем элементарные графические примитивы (точка, окружность и т. п.).

2. Возможность обучения нейронной сети позволяет строить модели, нечувствительные к воздействию шумов и искажений. Применение различных наборов зашумленных векторов позволило обучить сеть работать с изображениями, искаженными шумами, что характерно для реальной практики.

3. Использование пакета расширения Neural Network Toolbox дает возможность выполнять создание, обучение, моделирование, а также импорт и экспорт нейронных сетей и данных, используя только инструментальные возможности GUI-интерфейса.

4. Программная реализация процесса распознавания графических элементов с помощью системы Matlab не является сложной.

Литература

1.Фрактальный метод обнаружения групповых объектов транспортной инфраструктуры на изображениях / Е.П. Марков, А.С. Андрусенко, Е.И. Шабаков // Интеллектуальные технологии на транспорте. – 2016. – № 4(8). – С. 10–15.

2. Roushdy M. Detecting Coins with Different Radii based on Hough Transform in Noisy and Deformed Image / M. Roushdy // GVIP Journal, 2007, Vol. 7, No1. P. 1–5.

3. Хелгасон С. Преобразование Радона. – М. : Мир, 1983. – 152 с.

4. Смелянский Р. Л. Компьютерные сети. В 2 т. Том 2. Сети ЭВМ. – М. : Академия, 2011. – 240 с.

5. Epshtein B., Ofek E., Wexler Y. Detecting text in natural scenes with stroke width transform. In: CVPR ’10: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition (2010).

6. Principe J.C., Euliano N.R., Lefebvre W.C. Neural and Adaptive Systems. Fundamentals Through Simulations. – New York. John Wiley & Sons Inc. – 2000.

7. Luo F-L., Unbehauen R. Applied Neural Networks for Signal Processing. Cambridge University Press. – 2008.

8. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992–2000.

9. Баженов Р.И. Интеллектуальные информационные технологии. – Биробиджан : ПГУ им. Шолом-Алейхема, 2013. – 176 с.

10. Иванников В., Ланнэ А. Matlab для DSP. Нейронные сети: графический интерфейс пользователя [Электронный ресурс]. – URL: http://www.chipinfo.ru/ literature/chipnews/ 200108/1.html#lanne8.

11. Миронов И.С., Скурлаев С.В. Распознавание образов при помощи нейронной сети [Электронный ресурс]. – URL: http://confonline.susu.ru/index.php?option=com_content&view=article&id=57:2011-05-06-04-36-21&catid=16:-2----&Itemid=18.

12. Шеремет А.И., Перепелица В.В., Денисова А.М. Проектирование нейронной сети для распознавания символов в программной среде MATLAB [Электронный ресурс]. – URL: http://nauka.zinet.info/13/sheremet.php.

Identification of Images Using Neural Network Training

A. V. Krasnovidov, A. S. Alekseev
Emperor Alexander I St. Petersburg State Transport University
St.-Petersburg, Russia
alexkrasnovidow@mail.ru, affectum@hotmail.com

Abstract. **Computer vision and pattern recognition systems are used to solve a wide range of management, alert, military and medical purposes. Such systems perform the transformation of the original multidimensional data (color raster images) into a set of generalized output data (object classes, hypotheses, membership functions.). At the initial stages of this transformation, you often need to find graphics tools – circles, ellipses, rectangles, curves, lines, etc. – on a complex noisy image and build a map of graphic primitives. In the article various ways of allocation of such expectations are considered, their advantages and disadvantages are analyzed. The advantages of using trained neural networks for solving graphic primitive allocation problems are shown. With the help of the Matlab system, a model of the neural network is constructed to recognize the 26th Latin alphabet. The working capacity of the constructed model was checked.**

Keywords: **Neural networks, model of graphic primitives, Matlab, NeuralNetworkToolbox.**

REFERENCES

1. Fractal method of detecting group objects of transport infrastructure on images [Fraktal'nyy metod obnaruzheniya gruppovykh ob"ektov transportnoy infrastruktury na izobrazheniyakh] / E.P. Markov, A.S. Andrusenko, E.I. Shabakov // Intellectual Technologies on Transport [Intellektual'nye tekhnologii na transporte], 2016. – No 4 (8). – Pp. 10–15.

2. M. Roushdy. Detecting Coins with Different Radii based on Hough Transform in Noisy and Deformed Image / M. Roushdy // GVIP Journal, 2007, Vol. 7, No 1. – P. 1–5.

3. Helgason S. Transformation of Radon [Preobrazovanie Radona]. – Moscow : Mir, 1983, 152 p.

4. Smelyansky RL Computer networks [Komp'yuternye seti]. In 2 volumes. Volume 2. Computer networks; Academy – Moscow, 2011. – 240 c.

5. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: CVPR '10: Proc. of the 2010 Conference on Computer Vision and Pattern Recognition. (2010)

6. Principe J.C., Euliano N.R., Lefebvre W.C. Neural and Adaptive Systems. Fundamentals Through Simulations. New York. John Wiley & Sons Inc. 2000.

7. Luo F-L., Unbehauen R. Applied Neural Networks for Signal Processing. Cambridge University Press. – 2008.

8. Demuth H., Beale M. Neural Network Toolbox. For Use with MATLAB. The MathWorks Inc. 1992–2000.

9. Bazhenov R.I. Intellectual information technologies [Intellektual'nye informatsionnye tekhnologii]. Birobidzhan: PSU them. Sholom-Aleikhema, 2013. – 176 p.

10. Ivannikov V., Lanne A. Matlab for the DSP. Neural networks: graphical user interface [Matlab dlya DSP. Neyronnye seti: graficheskiy interfeys pol'zovatelya]. Available at: http://www.chipinfo.com/literature/chipnews/200108/1.html#lanne8 (accessed 25 May 2018).

11. Mironov I.S, Skurlaev S.V. Identification of images using a neural network [Raspoznavanie obrazov pri pomoshchi neyronnoy seti]. Available at: http://confonline.susu.ru/index.php?option=com_content&view=article&id=57:2011-05-06-04-36-21&catid=16:-2----&Itemid=18 (accessed 25 May 2018).

12. Sheremet AI, Perepelitsa VV, Denisova A.M. Designing a neural network for character recognition in the MATLAB software environment [Proektirovanie neyronnoy seti dlya raspoznavaniya simvolov v programmnoy srede MATLAB]. Available at:http://nauka.zinet.info/13/sheremet.php (accessed 25 May 2018).

Методика применения нечётких множеств в системе поддержки принятия решений робототехнического комплекса

С.В. Войцеховский, У.Ю. Головчанская, С.В. Логашев
ВКА имени А.Ф. Можайского,
Санкт-Петербург, РФ
vsv25@mail.ru, uljana18@gmail.com, loga1977@yandex.ru

Ю.С. Фоменко
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, РФ
kosmonavt.98@mail.ru

***Аннотация*. Предлагается методика использования математического аппарата нечеткого вывода на основе алгоритма Мамдани для системы поддержки принятия решений автономного робототехнического комплекса с целью повышения эффективности систем охраны государственных и ведомственных наземных объектов. Приводятся примерные виды входных и выходных функций принадлежности, варианты правил, результаты аккумулирования всех правил и дефазификации, пример вербально-числовой шкалы системы поддержки принятия решения, включающей варианты управляющих решений и контрмер в случае обнаружения злоумышленника.**

***Ключевые слова*: система поддержки принятия решений, потенциальный злоумышленник, нечеткий вывод, функции принадлежности, робототехнический комплекс.**

Введение

В настоящее время практически все разрабатываемые и принятые на снабжение наземные робототехнические комплексы (РК) используются в режиме дистанционного управления. Опыт применения таких роботов в локальных конфликтах последних лет показал их невысокую эффективность. Несомненно, дальнейшее развитие военной и специальной мобильной робототехники связано с повышением автономности наземных РК за счет передачи функций, выполняемых человеком-оператором, бортовым средствам, в основе работы которых лежат системы искусственного интеллекта или системы поддержки и принятия решений (СППР).

Некоторые элементы автономности, такие как запоминание маршрута и возврат в точку потери связи, обход препятствий по данным бортовых систем при супервизорном управлении и ряд других экспериментально проверены на макетных образцах и уже закладываются в возможности разрабатываемых роботов. Однако для осуществления полноценной автономности необходимо решить ряд проблем, о которых рассказывается в работе [1]. Одной из них является проблема автономного применения целевой нагрузки, включая разработку алгоритмов автоматического обнаружения, распознавания типа цели, «узнавания» конкретной цели, наведения оружия или иных средств.

Речь идёт о разработке системы искусственного интеллекта РК, которая способна автономно распознавать различные объекты и принимать решения в зависимости от складывающейся обстановки. Во многих странах, в том числе в РФ, разработаны и внедряются государственные программы создания перспективных РК (на период до 2030 года), неотъемлемой частью которых является их программное обеспечение. Актуальность работ в этой области обусловлена тем, что имеющийся значительный задел в области фундаментальных и поисковых исследований по различным проблемам искусственного интеллекта пока недостаточно реализован в реальных разработках [2].

В статье представлена методика применения аппарата нечеткого вывода в СППР автономного робототехнического комплекса (АРК) для повышения эффективности систем охраны наземных объектов МО РФ, ОАО «РЖД» и других ведомств. Для создания СППР АРК использовался математический аппарат нечеткого вывода на основе алгоритма Мамдани.

Применение алгоритма нечеткого вывода Мамдани

В настоящее время область применения нечеткой логики весьма обширна. Известно, что модель на основе нечеткого логического вывода прозрачнее (проще для понимания), чем аналогичные модели на дифференциальных, разностных или иных уравнениях [3].

При построении системы искусственного интеллекта могут использоваться различные подходы на основе нечеткого вывода или нейронные сети. В последние годы широкую известность получили подходы с использованием математического аппарата нечеткого вывода на основе алгоритма Мамдани [4]. Так, в работах [5–14] для достижения различных целей (повышения защищенности, устранения семантических противоречий, повышения устойчивости функционирования систем, оценки состояния безопасности информационных ресурсов и др.) применялся данный математический аппарат. Однако для расчетов в СППР АРК этот математический аппарат ранее не применялся, во всяком случае в открытых источниках авторы данной статьи его не встречали.

С точки зрения характеристической функции нечеткие множества являются естественным обобщением обычных множеств, когда мы отказываемся от бинарного характера этой функции и предполагаем, что она может принимать любые значения из отрезка [0, 1]. В теории нечетких множеств характеристическая функция называется функцией принадлежности [15].

Обобщенная модель функционирования СППР АРК

Принцип действия СППР заключается в непрерывном наблюдении АРК за обстановкой на охраняемом объекте. При этом АРК осуществляет обнаружение потенциальных злоумышленников до их проникновения на охраняемую территорию, осуществляя патрулирование по периметру охраняемого объекта на удалении 1–3 км. Энергоснабжение аккумуляторных батарей АРК планируется осуществлять в определенных точках, расположенных по маршруту движения АРК.

Действия по мониторингу выполняют сенсоры робототехнического комплекса. Данные с сенсоров поступают в модуль принятия решения, он, в свою очередь, опираясь на базу правил нечёткого вывода СППР, определяет степень уверенности в наличии потенциального злоумышленника и в зависимости от этого принимает соответствующее управляющее решение.

Модуль управления СППР предназначен для согласованного управления работой всех компонентов программного комплекса: запуск сенсоров робототехнического комплекса, портов и анализа журнала событий, редактирования базы правил, сбора информации от сенсоров и передачи этой информации в модуль принятия решения, который рассчитывает степень уверенности в наличии потенциальных злоумышленников.

Вся собранная модулем управления информация от сенсоров (звукового, оптического и дальномера) передаётся в модуль принятия решения, который на основе базы правил нечёткого вывода СППР определяет степень уверенности в наличии потенциального злоумышленника, в зависимости от этого принимает соответствующее управляющее решение (см. таблицу) и сообщает его через модуль связи начальнику караула охраняемого объекта.

Методика применения нечётких множеств

Дано

Входные данные с сенсоров РК (u_j, z_j, g_j, l_j, m_j, p_j),
где u_j – переменная для обнаружения невооруженных людей;

z_j – переменная для обнаружения раненных или убитых людей;

g_j – переменная для обнаружения вооруженных людей;

l_j – переменная для определения времени суток;

m_j – переменная для обнаружения выстрелов из огнестрельного оружия;

p_j – переменная для обнаружения наличия транспортных средств.

Примеры входных функций принадлежности (ФП) представлены на рисунке 1.

Лингвистическую переменную определяют как переменную, значениями (термами) которой являются не числа, а слова или предложения естественного языка. Каждому терму лингвистической переменной соответствует определенное нечеткое множество со своей функцией принадлежности, которая описывает совместимость этого терма с различными числовыми значениями [15].

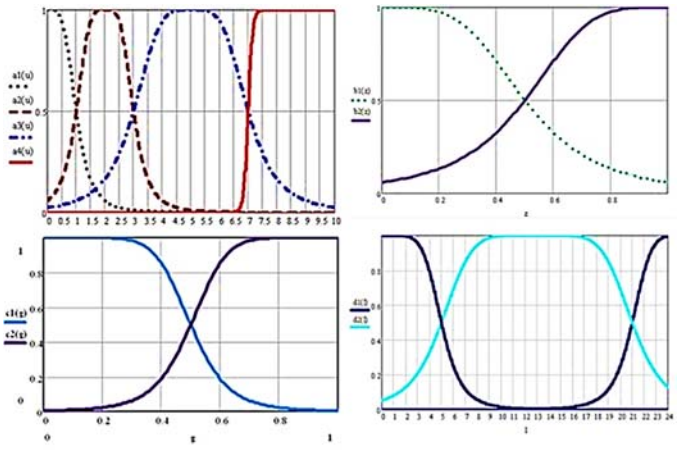


Рис. 1. Примеры входных функций принадлежности

Выходные данные: 1) y_i – переменная функции $G_i(y_i)$, предназначенная для определения степени уверенности в обнаружении потенциального злоумышленника до его проникновения на территорию охраняемого объекта (рис. 2);

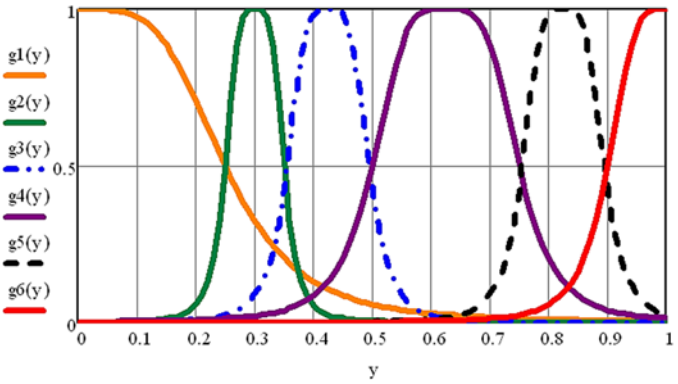


Рис. 2. График выходной функции $G(y)$

2) известные выражения для математического задания термов лингвистических переменных нечетких множеств:

$$A_i\left(u_j\right) = 1\Big/\left(1+\left(\frac{u_j - b_i}{a_i}\right)^{2s}\right); \quad (1)$$

$$A_i\left(u_j\right) = 1\Big/\left(1+e^{-s\left(u_j - h_i\right)}\right), \quad (2)$$

где a_i – половина λ-сечения нечёткого терм-множества; b_i – координата максимума терм-множества (середина ядра терм-множества); u_j – входные данные для переменной A; j – порядковый номер переменной; s – коэффициент ядра терм-множества (коэффициент лингвистического модификатора).

$$K_n = \min(A_i\left(u_j\right), B_i\left(z_j\right), C_i\left(g_j\right), D_i\left(l_j\right), E_i\left(m_j\right), F_i\left(p_j\right)), \quad (3)$$

где K_n – агрегированная степень истинности предпосылок правила n.

$$P_n(y) = \min(K_n, G_i(y)), \quad (4)$$

где $P_n(y)$ – модифицированная ФП этих заключений для каждого правила на основе выполнения композиционной операции, модифицированной для нечёткой продукции, между определённым на предыдущем этапе агрегированным значением степеней истинности предпосылок правила, например K_n, и соответствующей ФП его заключения $G_i(y)$.

Для получения данных о номере правила, имеющего наибольшую активизированную степень истинности из всех правил, необходимо сравнить агрегированные степени истинности предпосылок правил (полученных на предыдущем этапе) и найти максимальную из них по формуле:

$$K_{max} = \max(K_n),\ n = \overline{1,l}, \quad (5)$$

где l – количество правил в базе.

Для каждой из выходных ФП производится объединение нечетких множеств и формируется одно нечеткое множество $F_{rez}(y)$. Результат аккумулирования для ФП, активизированных на предыдущем этапе, находится путём объединения соответствующих нечётких множеств по формуле (6) с использованием операции max-дизъюнкции:

$$F_{rez}(y) = \max\left(P_1(y), P_2(y), \cdots P_l(y)\right). \quad (6)$$

Требуется

С помощью формул (1)–(7) осуществить расчёт модифицированного значения Y – степени уверенности СППР в наличии потенциального злоумышленника до его проникновения на охраняемую территорию. Результат дефаззификации: чёткое значение Y выходной переменной y рассчитывается как центр тяжести ФП $F_{rez}(y)$ по формуле:

$$Y = \int_0^1 y F_{rez}(y)\,dy \Big/ \int_0^1 F_{rez}(y)\,dy, \quad (7)$$

где 0, 1 – границы интервала носителя нечёткого множества выходной лингвистической переменной y.

1. Формирование нечетких рассуждений с использованием алгоритма Мамдани включает следующие этапы [3; 6; 16; 17]: введение нечёткости; агрегирование степеней истинности предпосылок правил; активизация заключений нечётких продукционных правил; аккумулирование результатов всех правил; приведение к чёткости (дефаззификация); принятие управляющих решений и контрмер.

Эксперты самостоятельно оценивают вывод по каждому правилу. На основе согласованных экспертных мнений формируется база правил нечеткого вывода СППР. При ее формировании с помощью экспертов будем использовать подход, представленный в работе [6]. Таким образом, правило № 1 СППР примет следующий вид: ЕСЛИ u_1 есть $A_i(u_j)$ И z_1 есть $B_i(z_j)$ И g_1 есть $C_i(g_j)$ И l_1 есть $D_i(l_j)$ И m_1 есть $E_i(m_j)$ И p_1 есть $F_i(p_j)$, ТО y_1 есть $G_i(y_j)$. Более подробно вопросы создания базы правил нечёткого вывода рассматриваются в работах [5; 6].

ВВЕДЕНИЕ НЕЧЁТКОСТИ

Допустим, датчиками СППР получены следующие входные данные: u = 1, z = 0, g = 0, l = 12, m = 0, p = 0. Эти входные данные подставим в каждое правило СППР и определим степени истинности каждой предпосылки каждого правила.

Например, подставляя в ФП входные данные, указанные в правиле № 1, в соответствии с типом формул (1), (2) и другими параметрами получим следующие значения степеней истинности его предпосылок: $A_1(1)$=0,5; $B_1(0)$=1, $C_1(0)$=1, $D_1(12)$=0,016; $E_1(0)$=1, $F_1(0)$=1. Переведя результаты в проценты (округляем результаты до десятых), получим: $A_1(1)$=50; $B_1(0)$=100, $C_1(0)$=100, $D_1(12)$=1,6; $E_1(0)$=100, $F_1(0)$=100. Аналогично для правила № 2 получим следующие значения степеней истинности его предпосылок:
$A_1(1)$=0,5; $B_1(0)$=1, $C_1(0)$=1, $D_1(12)$=0,016; $E_1(0)$=1, $F_2(0)$=0,004.

Аналогично определим значения степеней истинности остальных правил.

2. Агрегирование степеней истинности предпосылок правил, объединённых с помощью нечёткой логической операции конъюнкции (И), осуществляется по формуле min-конъюнкции (3) [5; 6; 9; 16].

Выполним агрегирование степени истинности предпосылок по каждому из нечетких продукционных правил СППР. Тогда агрегированная степень истинности предпосылок наличия потенциального злоумышленника по правилу № 1 рассчитывается с помощью формулы:

$$K_1 = \min(A_1(u), B_1(z), C_1(g), D_1(l), E_1(m), F_1(p)),$$

где K_1 – агрегированная степень истинности предпосылок правила № 1;

$A_1(u)$, $B_1(z)$, $C_1(g)$, $D_1(l)$, $E_1(m)$, $F_1(p)$ – значения ФП предпосылок правила № 1 (полученные на этапе 1).

Тогда K_1 = min(0,5; 1; 1; 0,016; 1; 1)=0,016. Для правила № 2 имеем K_2=min(0,5; 1; 1; 0,016; 0,004) и т. д.

В работах [5; 9; 15] рассматриваются системы нечеткого вывода, в которых вводится пороговое ограничение для каждого правила ($K_n \geq 0{,}05$), для того чтобы не активизировались правила, которые являются малозначительными. Это приводит к снижению сложности алгоритма и не оказывает существенного влияния на точность расчёта степени уверенности СППР в наличии потенциального злоумышленника.

3. Процедура активизации (композиция или определение степеней истинности) заключений нечётких продукционных правил состоит в определении модифицированных ФП этих заключений для каждого правила на основе выполнения композиционной операции, модифицированной для нечёткой продукции, между определённым на предыдущем этапе агрегированным значением степеней истинности предпосылок правила, например K_n, и соответствующей ФП его заключения $G_i(y)$. Для правила № 1 формула примет вид:

$$P_1(y) = \min(K_1, G_1(y)).$$

Активизированная ФП $P_3(y)$ для правила № 3 ($K_n \geq 0{,}05$) приведена на рисунке 3.

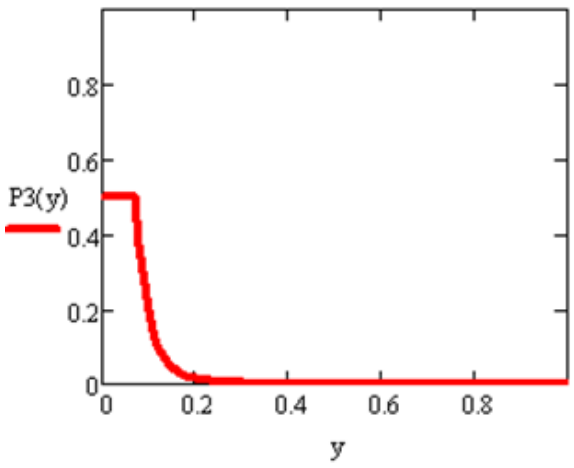


Рис. 3. Активизированная ФП $P_3(y)$

Необходимо выявить номер правила n, которое и определяет результат нечёткого вывода (обычно существует одно такое правило), сохранить номер правила в журнал событий СППР в строку соответствующих входных данных (для последующего анализа в случае ошибки).

4. Аккумулирование результатов всех правил.

На рисунке 4 показаны заключения нечётких продукционных правил, агрегированные степени истинности которых удовлетворяют условию $K_n \geq 0{,}05$, а на рисунке 5 – результат их аккумулирования.

Далее выполняется аккумулирование (накопление) результатов всех остальных правил. Для каждой из выходных ФП производится объединение нечетких множеств и формируется одно нечеткое множество $F_{rez}(y)$, при этом результаты тех правил, где получено значение 0, не учитываются.

Результат аккумулирования для ФП, активизированных на этапе 3, находится путём объединения соответствующих нечётких множеств по формуле (6) с использованием операции max-дизъюнкции. На рисунке 5 показан результат аккумулирования нечётких продукционных правил.

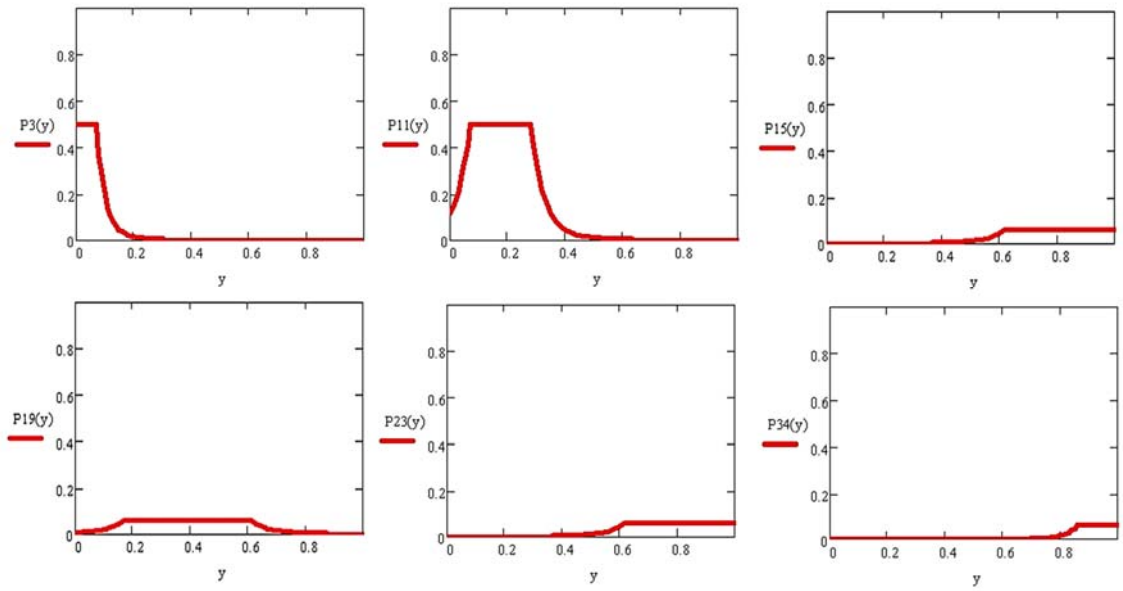


Рис. 4. Аккумулируемые функции принадлежности (6 правил)

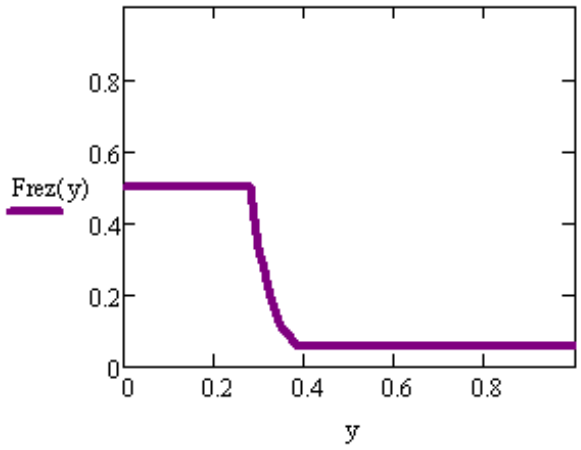


Рис. 5. Результат аккумулирования всех правил

5. Приведение к чёткости (дефаззификация) используется, чтобы преобразовать нечёткий набор выводов в чёткое число. Применяемые в современных системах управления устройства и механизмы способны воспринимать традиционные команды в форме количественных значений соответствующих управляющих переменных. Именно по этой причине необходимо преобразовать нечеткие множества в некоторые конкретные значения переменных. Для СППР приведение к чёткости – это нахождение степени уверенности в наличии потенциального злоумышленника Y.

Для выполнения численных расчетов на этапе дефаззификации используется *метод центра тяжести* – нахождение абсциссы центра тяжести площади, ограниченной графиком кривой функции принадлежности соответствующей выходной переменной. Результат дефаззи-

фикации – чёткое значение *Y* выходной переменной у рассчитывается как центр тяжести ФП $F_{rez(y)}$ по формуле (7). При подстановке указанных выше входных данных в (7) получим: *Y*=0,26 (рис. 6).

6. Принятие управляющих решений и контрмер.

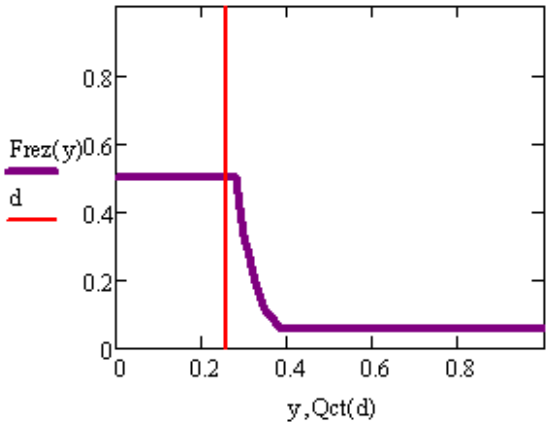


Рис. 6. Результат дефаззификации по методу центра тяжести

Для оценки степени уверенности в наличии потенциального злоумышленника по числовому значению, полученному по формуле (7), предлагается использовать вербально-числовую шкалу [5]. В состав вербально-числовых шкал, как правило, входят содержательное описание градаций шкалы и числовые значения, соответствующие каждой из градаций шкалы. Используем в качестве вербально-числовой шкалы, имеющей достаточно широкое применение, шкалу Харрингтона, характеризующую степень выраженности критериального свойства и имеющую универсальный характер (см. таблицу).

СППР должна еще и своевременно среагировать на злоумышленника. Дополним названную шкалу управляющим решением и контрмерами, которые принимаются модулем принятия решения СППР.

Получив значение степени уверенности наличия потенциального злоумышленника 0,26 (см. рис. 6) – «низкая», в соответствии с пунктом 5 таблицы СППР самостоятельно принимает решение: «Вероятность наличия потенциального злоумышленника низкая. Переместиться в квадрат F для дальнейшего обзора местности».

ТАБЛИЦА. Вербально-числовая шкала СППР

№ п/п	Лингвистическая градация наличия злоумышленника	Значение *Y*	Варианты управляющих решений и контрмер
1	Очень высокая	0,9–1,0	Обнаружен потенциальный злоумышленник. Включить оповещающую сирену. Вызвать дежурную смену охраны. Сообщить координаты местонахождения потенциального злоумышленника
2	Высокая	0,75–0,9	Вероятность наличия потенциального злоумышленника высока. Сделать фото обнаруженного объекта и отправить на пункт охраны. Вызвать досмотровую группу. Остаться на месте. Быть в готовности дать целеуказания о местонахождении потенциального злоумышленника
3	Выше среднего	0,5–0,75	Вероятность наличия потенциального злоумышленника высока. Сделать фото обнаруженного объекта и отправить его на пункт охраны. Вызвать досмотровую группу
4	Ниже среднего	0,35–0,5	Вероятность наличия потенциального злоумышленника невысокая. Сделать фото обнаруженного объекта и отправить его на пункт охраны. Продолжить наблюдение в данном квадрате
5	Низкая	0,25–0.35	Вероятность наличия потенциального злоумышленника низкая. Переместиться в квадрат F для дальнейшего обзора местности
6	Очень низкая	0,0–0,25	Ничего не обнаружено. Продолжить наблюдение

Заключение

В настоящее время практически все разрабатываемые и принятые на снабжение наземные РК используются в режиме дистанционного управления и управляются человеком-оператором. Для автономных робототехнических комплексов актуальной является задача разработки системы искусственного интеллекта РК, которая способна автономно распознавать различные объекты и принимать решения в зависимости от складывающейся обстановки. Предложенная методика применения математического аппарата нечётких множеств в СППР позволяет автоматически (в реальном масштабе времени) определять степень уверенности АРК в наличии потенциального злоумышленника, принимать соответствующее управляющее решение, контрмеры и таким образом повысить эффективность систем охраны государственных и ведомственных наземных объектов МО РФ и ОАО «РЖД».

Литература

1. Лапшов В.С. Перспективы разработки автономных наземных робототехнических комплексов специального военного назначения / В.С. Лапшов, В.П. Носков и др. // Известия ЮФУ. Технические науки. – 2016. – № 1 (174). – С. 156–168.

2. Лопота А.В. Наземные робототехнические комплексы военного и специального назначения [Электронный ресурс] / А.В. Лопота, А.Б. Николаев // rtc.ru: сайт ЦНИИ РТК. – Режим доступа: http://rtc.ru/images/docs/book/nazemnie.pdf. – Заглавие с экрана. – (Дата обращения: 25.03.2018).

3. Штовба С.Д. Введение в теорию нечетких множеств и нечеткую логику / С.Д. Штовба. – Винница : Изд. Винницкого ГТУ, 2001. – 198 с. – Режим доступа: matlab.exponenta.ru. (Дата обращения: 13.07.2017).

4. Mamdani E.H. Application of fuzzy logic to approximate reasoning using linguistic Systems // Fuzzy Sets and Systems, 1977. – V. 26. – P. 1182–1191.

5. Войцеховский С.В. Выявление вредоносных программных воздействий на основе нечеткого вывода / С.В. Войцеховский, А.Д. Хомоненко // Проблемы информационной безопасности. Компьютерные системы. – 2011. – № 3. – С. 81–91.

6. Войцеховский С.В. Согласование экспертных оценок при нечётком выводе в системе обнаружения вторжений / С.В. Войцеховский, А.Д. Хомоненко // Проблемы информационной безопасности. Компьютерные системы. – 2009. – № 4. – С. 42–50.

7. Mamdani E.H. and Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 1975. – № 7. – P. 1–13.

8. Uziel Sandler, Lev Tsitolovsky Neural Cell Behavior and Fuzzy Logic. Springer, 2008. – 478 c.

9. Войцеховский С.В. Методика повышения устойчивости функционирования военных систем на основе нечеткой логики / С.В. Войцеховский, С.В. Калиниченко и др. // Известия Тульского государственного университета. Технические науки. – 2017. – № 9–1. – С. 450–458.

10. Корченко А.Г. Построение систем защиты информации на нечётких множествах. Теория и практические решения / А.Г. Корченко – К. : МК-Пресс, 2006 – 320 с.

11. Климкина А.А. Выбор модели SSD-накопителя на основе алгоритма нечёткого вывода / А.А. Климкина, А.О. Валиева и др. // Интеллектуальные технологии на транспорте. – 2017. – № 3 (11). – С. 32–38.

12. O. Obe, and I. Dumitrache. Fuzzy control of autonomous mobile robot. U.P.B. Sci. Bull., Series C, Vol. 72, Iss. 3, 2010.

13. V. Mayoral, A. Hernández, R. Kojcev, I. Muguruza, I. Zamalloa, A. Bilbao, and L. Usategi. The shift in the robotics paradigm; the hardware robot operating system (h-ros); an infrastructure to create interoperable robot components. In 2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), July 2017, pp. 229–236.

14. Alonso, J.M., Castiello, C., Mencar, C.: The role of interpretable fuzzy systems in designing cognitive cities. In Portmann, E., Seising, R., Tabachi, M., eds.: Designing Cognitive Cities: Linking Citizens to Computational Intelligence to Make Efficient, Sustainable and Resilient Cities a Reality. Studies in Systems, Decision and Control. Springer Verlag (2017) 1-21.

15. Жирабок А.Н. Нечеткие множества и их использование для принятия решений / А.Н. Жирабок // Соросовский образовательный журнал. – 2001. – № 2. – С. 109–115.

16. Борисов В.В. Нечеткие модели и сети / В.В. Борисов, В.В. Круглов, А.С. Федулов – М. : Горячая линия-Телеком, 2007. – 284 с.

17. Круглов В.В. Нечеткая логика и искусственные нейронные сети : учеб. пособие / В.В. Круглов, М.И. Дли, Р.Ю. Голунов – М. : Изд-во физ.-мат. лит-ры, 2001. – 224 с.

Method of Application of Fuzzy Sets in the System of Support of Decision-Making of the Robotechnical Complex

S.V. Voytsekhovsky, U.Yu. Golovchanskaya, S.V. Logashov
A.F. Mozhaisky Military Aerospace Academy
St. Petersburg, Russia
vsv25@mail.ru, uljana18@gmail.com, loga1977@yandex.ru

Yu.S. Fomenko
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
kosmonavt.98@mail.ru

***Abstract*. The technique of use of a mathematical apparatus of an indistinct conclusion on the basis of an algorithm Mamdani for the system of support of decision-making of an autonomous robotic complex for the purpose of increase in efficiency of systems of protection of the state and departmental land objects is offered. Approximate types of entrance and output functions of accessory, versions of rules, results of accumulation of all rules and a defuzzification, an example of a verbal and numerical scale of system of support of decision-making, including versions of the operating decisions and counter-measures in case of detection of the malefactor are given.**

***Keywords*: decision support system, potential attacker, fuzzy conclusion, membership functions, robotic complex.**

References

1. Lapshov VS Prospects for the development of autonomous terrestrial robotic complexes of special military purpose / V.S. Lapshov, V.P. Noskov et al. // Izvestia of SFedU. Technical science. – 2016, № 1 (174), pp. 156–168.

2. Lopota A.V. Ground-based robotic complexes for military and special purposes [Electronic resource] / A.B. Lopota, A.B. Nikolaev // rtc.ru: site CRI RTC. – Access mode: http://rtc.ru/images/docs/book/nazemnie.pdf. – The title from the screen (accessed 25.03.2018).

3. Shtovba S.D. Introduction to the theory of fuzzy sets and fuzzy logic. / S.D. Shtovba. – Vinnytsia: Ed. Vinnitsa State Technical University, 2001. 198 pp. – Access mode: matlab.exponenta.ru (accessed July 13, 2017).

4. Mamdani E.H. Application of fuzzy logic to approximate reasoning using linguistic Systems // Fuzzy Sets and Systems, 1977, V. 26, pp. 1182–1191.

5. Voytsekhovsky S.V. Identification of malicious software actions based on fuzzy inference / S.V. Voytsekhovsky, A.D. Homonenko // Problems of Information Security. Computer systems – St. Petersburg., 2011, № 3, pp. 81–91.

6. Voytsekhovsky S.V. Harmonization of expert assessments with fuzzy inference in the intrusion detection system / S.V. Voytsekhovsky, A.D. Homonenko // Problems of Information Security. Computer systems – St. Petersburg, 2009, № 4, pp. 42–50.

7. Mamdani E.H. and Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, 1975, № 7, pp. 1–13.

8. Uziel Sandler, Lev Tsitolovsky Neural Cell Behavior and Fuzzy Logic. Springer, 2008, 478 p.

9. Voytsekhovsky S.V. Technique of increasing the stability of military systems on the basis of fuzzy logic / S.V. Voytsekhovsky, S.V. Kalinichenko et al. // Izvestiya of the Tula State University. Technical science. 2017. № 9–1, pp. 450–458.

10. Korchenko A.G. Construction of information security systems on fuzzy sets. Theory and practical solutions. / A.G. Korchenko-K: MK-Press, 2006, 320 p.

11. Klimkina A.A. Choice of model SSD-drive on the basis of algorithm of fuzzy inference / A.A. Klimkina, A.O. Valieva et al. // Intellectual Technologies in Transport. 2017, № 3 (11), pp. 32–38.

12. O. Obe, and I. Dumitrache. Fuzzy control of autonomous mobile robot. U.P.B. Sci. Bull., Series C, Vol. 72, Iss. 3, 2010.

13. V. Mayoral, A. Hernández, R. Kojcev, I. Muguruza, I. Zamalloa, A. Bilbao, and L. Usategi. The shift in the robotics paradigm; the hardware robot operating system (h-ros); an infrastructure to create interoperable robot components. In 2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), July 2017, pp. 229–236.

14. Alonso, J.M., Castiello, C., Mencar, C.: The role of interpretable fuzzy systems in designing cognitive cities. In Portmann, E., Seising, R., Tabachi, M., eds.: Designing Cognitive Cities: Linking Citizens to Computational Intelligence to Make Efficient, Sustainable and Resilient Cities a Reality. Studies in Systems, Decision and Control. Springer Verlag (2017) 1–21.

15. Zhirabok A.N. Fuzzy sets and their use for decision making / A.N. Zhiraubok // Soros Educational Journal. 2001, № 2, pp. 109-115.

16. Borisov V.V. Fuzzy models and networks. / V.V. Borisov, V.V. Kruglov, A.S. Fedulov - M .: Hot line-Telecom, 2007, 284 p.

17. Kruglov V.V. Fuzzy logic and artificial neural networks: Proc. allowance. / V.V. Kruglov, M.I. Long, R.Yu. Golunov – Moscow: Publishing House of Physical and Mathematical Literature, 2001, 224 p.

Сравнительный анализ алгоритмов распределения работ в мультипроцессорных системах

И.А. Молодкин, С.Г. Свистунов
Петербургский государственный университет путей сообщения
Императора Александра I
Санкт-Петербург, Россия
imolodkin@gmail.com, ssg47@mail.ru

Аннотация. **Приводятся описания алгоритма Дегтярёва и генетического алгоритма, применяемых для распределения работ в мультипроцессорных вычислительных системах, а также результаты сравнительного анализа качества распределения алгоритмов путем сравнения общего времени работы системы после распределения работ. Анализ проведен с помощью разработанной программной имитационной модели. Показано, что алгоритм Дегтярёва обеспечивает быстрый результат с хорошими характеристиками при краткосрочном планировании. Он прост в реализации и работает сравнительно быстрее генетического алгоритма. Преимуществом генетического алгоритма можно считать лучшее распределение работ при их малом количестве и достаточном времени на вычисления. Сформулированы рекомендации по применению алгоритмов при разных условиях.**

Ключевые слова: **мультипроцессорные системы, алгоритмы, генетические алгоритмы, алгоритм Дегтярёва.**

Введение

В современном мире все большее распространение получают различные мультипроцессорные системы, которые призваны ускорить процесс решения сложных задач. Ведущие европейские, американские и китайские компании по разработке таких машин инвестируют сотни миллионов долларов в развитие этой отрасли [1]. Инженерам приходится решать разнообразные проблемы, такие как обеспечение масштабируемости, обеспечение общего доступа к памяти и др. [2].

Важной задачей при построении мультипроцессорных систем является балансирование нагрузки на вычислительные узлы (процессоры). Это необходимо для оптимального режима работы системы в целом, минимизации времени простоя и максимальной производительности. При решении этой задачи применяются различные алгоритмы из области теории расписаний [3].

Цель работы – проанализировать характеристики двух алгоритмов для составления оптимального плана распределения работ в мультипроцессорной системе. Первый алгоритм предложен профессором Ю.И. Дегтярёвым [4] в книге «Исследование операций» [5]. Второй – генетический алгоритм [6], оптимизированный для данной задачи. Оба этих алгоритма позволяют получить приближенное к оптимальному распределение работ.

Для проведения анализа на языке Java написана программная модель мультипроцессорной системы, использующая указанные алгоритмы для распределения работ. Интерфейс представлен на рис. 1.

Постановка задачи

Пусть имеется N работ и L процессоров, предназначенных для выполнения этих работ (предполагаем, что $N \ggg L$). Любая работа может быть проведена в любом канале, и время, необходимое для этого, известно. Требуется распределить указанные N работ по L процессорам так, чтобы полное время T_C занятости системы было минимальным.

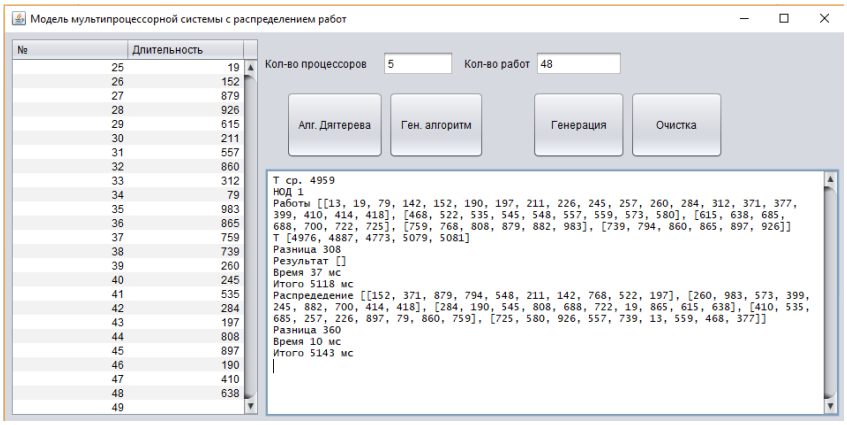


Рис. 1. Интерфейс программной имитационной модели

Очевидно, что полное время занятости системы будет определяться максимальным временем занятости процессоров.

Алгоритм Дегтярёва

Как пишет Ю. И. Дегтярёв [4], алгоритм предполагает последовательные коррекции схем загрузки процессоров начиная с некоторой предварительной схемы, получаемой эвристическом путем.

Время занятости процессора (канала) с номером l ($1 \le l \le L$) оценивается как

$$T_l = \sum_{v=1}^{n_i} (\tau_{vt} + \Delta t_{v-1,l}),$$

где n_l – количество проводимых в нем работ.

Пусть известные длительности работ $\tau_j \left(j = \overline{1,N} \right)$ упорядочены по признаку возрастания и соответственно пронумерованы. В таком случае легко определяется наименьший номер $j = p$, отвечающий условию

$$\sum_{j=1}^{p} \tau_j \ge T_{\text{ср}},$$

что позволяет сравнить между собой разности

$$\sum_{j=1}^{p} \tau_j - T_{\text{ср}}, \quad T_{\text{ср}} - \sum_{j=1}^{p-1} \tau_j$$

и выбрать в качестве T_l либо

$$\sum_{j=1}^{p} \tau_j$$

(если первая разность меньше второй), либо

$$\sum_{j=1}^{p-1} \tau_j$$

(если вторая разность меньше первой), исключив тем самым из дальнейшего рассмотрения τ_j, составившие T_l.

Повторяя эту несложную процедуру применительно к оставшимся τ_j, получаем сначала T_2, затем T_3 и т. д. Среди указанных T_1, T_2, находится величина maxT_l, так что

$$T_{\text{ср}} \le \min T_c \le \max T_l.$$

Если расхождение между $T_{\text{ср}}$ и найденным maxT_l окажется слишком большим, то возникнет необходимость анализа большого числа комбинаций T_1, T_2, … T_l, многие из которых будут затем отброшены как ненужные. Чтобы избежать этого, достаточно найти среди полученных T_1, T_2, … T_l величину minT_l, вычислить разность (maxT_l – minT_l) и перераспределить работы между наиболее и наименее загруженными каналами с целью получить лучшее (меньшее) значение maxT_l.

Поскольку трудоемкость подобных операций невелика, имеет смысл их повторять, добиваясь приемлемых отклонений maxT_l от $T_{\text{ср}}$.

После того как межканальный обмен работами будет завершен, станет возможным оценить погрешность результата, выражаемую либо разностью maxT_l – $T_{\text{ср}}$, либо отношением

$$\frac{\max T_l - T_{\text{ср}}}{T_{\text{ср}}}.$$

Если эта погрешность находится в допустимых (с точки зрения исследователя) пределах, то процесс оптимизации расписания для мультипроцессора заканчивается.

Генетический алгоритм

Генетический алгоритм часто применяется для решения задач оптимального распределения работ [6].

Идея генетических алгоритмов предложена Джоном Холландом в 1960-х годах, а результаты первых исследований обобщены в его монографии «Адаптация в природных и искусственных системах» [7], а также в диссертации его аспиранта Кеннета Де Йонга [8].

Генетический алгоритм использует для работы эволюционные принципы наследственности, изменчивости и естественного отбора и такие механизмы, как кроссинговер и мутация [9; 10]. Он работает с популяцией особей, в каждой из которых закодировано возможное решение задачи.

Одноточечный кроссинговер (Single-point crossover) осуществляется таким образом: выбирается случайная точка внутри особи, она называется точкой разрыва; в этой точке две особи делятся на части и обмениваются ими друг с другом.

После механизма кроссинговера запускается процедура мутации. Эта процедура позволяет выводить популяцию из локальных экстремумов и предотвращает раннюю сходимость. Ее суть довольно проста: случайным образом выбирается и изменяется ген в особи.

Вероятность мутаций гарантирует появление многообразия в популяции, но вполне возможно и разрушение хорошей особи при мутировании. Перьякс с соавторами предложили использовать метод супериндивидуального приближения для отбора особей в новую популяцию [11; 12].

При супериндивидуальном приближении выбирается наилучшая особь среди числа родителей и их потомков – элитная хромосома. Все хромосомы новой популяции являются копиями этой элитной хромосомы. Поэтому в популяции будут происходить сильные мутации.

Исходя из условий задачи механизм мутации в данной реализации неприменим.

Для того чтобы оценить качество закодированных решений, используют функцию приспособленности (фитнесс-функцию), которая необходима для вычисления приспособленности каждой особи.

Рассмотрим параметры алгоритма, который был применен для распределения работ в имитационной модели.

Формирование начальной популяции производится случайным образом в заданном заранее количестве n и примерно равно десятикратному числу работ. При этом работы распределяются по генам (процессорам) также случайным образом. Число поколений равно пяти.

В качестве значения функции приспособленности принимается значение разницы между минимальным и мак-

симальным временем работы процессора. Три особи с лучшим значением функции приспособленности подвергаются кроссинговеру.

Имитационная модель

Для проведения сравнительного анализа была разработана программная имитационная модель на языке высокого уровня Java.

В программной модели реализовано три класса:

- Multiproc;
- MainFrame;
- CustomOutputStream.

Класс Multiproc содержит методы, реализующие сами алгоритмы и методы взаимодействия с другими классами (ввод/вывод).

Класс MainFrame отвечает за графический интерфейс пользователя, ввод исходных данных и вывод результатов.

Класс CustomOutputStream является вспомогательным и обеспечивает вывод подробной информации в консоль.

В программе реализована проверка входных данных на корректность, что позволяет снизить вероятность ошибки и повысить удобство использования.

Программа содержит 3340 строк кода. Алгоритм Дегтярёва занимает почти 500 строк, генетический алгоритм – всего 150. Остальной код обеспечивает работу графического интерфейса и ввод/вывод.

Несмотря на большую разницу в объеме кода, алгоритм Дегтярёва является более быстрым и эффективнее использует оперативную память. Это связано с тем, что в генетическом алгоритме используется много ресурсов для хранения всех особей и работы с ними.

Анализ характеристик

Сравнительный анализ был разделен на два этапа и заключался в сравнении времени работы и качества распределения работ при разных условиях.

Первый этап анализа

На первом этапе проведем замеры общего времени работы системы T при распределении работ, длительность которых не превышает 10 с.

Оценим качество распределения путем сравнения общего времени работы системы, получаемого при помощи одного и другого алгоритма при увеличении числа работ и процессоров (табл. 1).

ТАБЛИЦА 1. Время работы системы при распределении больших работ и разном числе процессоров

N	P	T, мс		Разница, %
		Дегтярёва	Генетический	
25	4	318081	313796	1,37
100	10	97497	98781	–1,30
250	25	520744	552784	–5,80
375	30	662244	678416	–2,38
500	45	603077	687692	–12,30

Как видно из результатов, при малом числе работ разница между результатами работы алгоритмов не превышает 6 %, но при увеличении числа работ свыше 500 алгоритм Дегтярёва показывает лучшие результаты. Это связано со случайностью начального распределения в генетическом алгоритме.

Оценим влияние числа процессоров P на качество распределения (табл. 2). Число работ N примем равным пятидесяти.

ТАБЛИЦА 2. Время работы системы при распределении больших работ

P	T, мс		Разница, %
	Дегтярёва	Генетический	
5	559578	548729	2,0
15	192445	178979	7,5
25	152100	146248	4,0
35	106488	95417	11,6

Увеличение числа процессоров слабо влияет на результаты работы алгоритмов, однако при отношении P к N, большем 0,5, отмечается резкое увеличение разницы между результатами алгоритмов.

Анализ разницы распределения работ при увеличении числа работ N покажет, как изменяется характеристика алгоритмов при увеличении размеров распределяемых задач (табл. 3) при числе процессоров P, равном пяти.

ТАБЛИЦА 3. Время работы системы при распределении больших работ и фиксированном числе процессоров

N	T, мс		Разница, %
	Дегтярёва	Генетический	
50	562814	541170	4,0
100	1051102	1041090	1,0
300	3015444	3041735	–0,9
500	5050111	5139354	–1,7

По результатам тестирования при одинаковом числе процессоров и разном числе работ разница между характеристиками распределения составляла не более 4 %.

Второй этап анализа

Второй этап анализа проводился при распределении работ длительностями не более секунды. Это позволило оценить применение того или иного алгоритма при распределении коротких работ.

По аналогии с первым этапом оценим качество распределения путем сравнения общего времени работы системы, получаемого при помощи одного и другого алгоритма при увеличении числа работ и процессоров (табл. 4, рис. 2).

ТАБЛИЦА 4. Время работы системы при распределении небольших работ и разном числе процессоров

N	P	T, мс		Разница, %
		Дегтярёва	Генетический	
25	4	2908	2859	1,7
100	10	5738	5921	–3,1
250	25	5676	9402	–39,6
375	30	6778	8814	–23,1
500	45	5893	10753	–45,2

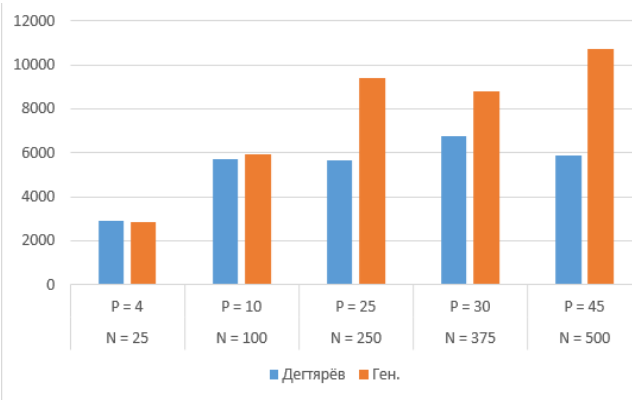


Рис. 2. Диаграмма на основе табл. 4

Исходя из результатов проведенных измерений в случае распределения коротких задач увеличение их числа негативно сказывается на качестве распределения с помощью генетического алгоритма. Он показывает результаты хуже на 23–45 %, чем результаты алгоритма Дегтярёва.

Оценим влияние числа процессоров на качество распределения (табл. 5).

ТАБЛИЦА 5. Время работы системы при распределении небольших работ и разном числе процессоров

P	T, мс		Разница, %
	Дегтярёва	Генетический	
5	5756	5781	–0,4
15	1907	1807	5,5
25	1396	1437	–2,9
35	928	938	–1,1

Как и в случае с длительными работами, увеличение числа процессоров несущественно влияет на качество распределения.

Проверим, какое влияние окажет увеличение числа работ при одинаковом числе процессоров (табл. 6).

ТАБЛИЦА 6. Время работы системы при распределении большего числа работ и фиксированном числе процессоров

N	T, мс		Разница, %
	Дегтярёва	Генетический	
50	4793	4878	–1,7
100	9176	9689	–5,3
300	29850	32153	–7,2
500	49844	61045	–18,3

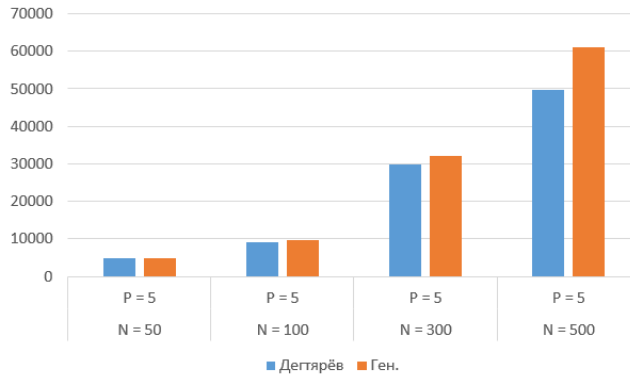


Рис. 3. Диаграмма на основе табл. 6

На рис. 3 четко виден тренд на ухудшение характеристик распределения генетическим алгоритмом по сравнению с алгоритмом Дегтярёва.

Итоги анализа

После проведения замеров характеристик распределения можно сделать следующие выводы.

1. Если время, требуемое для решения задач, составляет минуты или более, а число задач не превышает нескольких сотен, предпочтительнее использовать генетический алгоритм. При данных условиях он показывает лучшее распределение.
2. При распределении коротких по времени задач генетический алгоритм проигрывает алгоритму Дегтярёва по времени его выполнения.
3. Влияние числа процессоров на характеристики распределения можно считать несущественным.

Заключение

Мультипроцессорные системы прочно вошли во все сферы информационных технологий. Оптимизация их работы, в частности оптимальное распределение работ в их узлах, является важной задачей на данный момент, ведь от этого зависит общая производительность всей системы в целом [13; 14].

Для определения характеристик рассматриваемых алгоритмов была разработана имитационная модель мультипроцессорной системы. С ее помощью были проведены тесты алгоритмов при различных условиях и входных данных.

Проведенный анализ показал, что рассмотренные в статье алгоритмы с успехом могут применяться при различных режимах планирования. Алгоритм Дегтярёва обеспечивает быстрый результат с хорошими характеристиками при краткосрочном планировании. Он прост в реализации и работает сравнительно быстрее генетического алгоритма. Преимуществом генетического алгоритма можно считать лучшее распределение работ при их малом количестве и достаточном времени на вычисления.

Литература

1. Суперкомпьютеры_(мировой_рынок). – http://www.tadviser.ru/index.php/ (Дата обращения 25.05.2018).

2. Многопроцессорные ЭВМ и методы их проектирования [Текст] / Б.А. Бабаян, А.В. Бочаров, В.С. Волин и др. – М. : Высшая школа, 1990. – 384 с.

3. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы. – М. : МГУ им. Ломоносова, 2011. – 222 с.

4. Дегтярёв Юрий Иванович. – URL: https://mai.ru/content/people/ index.php?ID=28066 (Дата обращения 25.05.2018).

5. Дегтярёв Ю.И. Исследование операций. – М. : Высшая школа, 1986. – 320 с.

6. Панченко Т.В. Генетические алгоритмы [Текст] : учебно-методическое пособие / под ред. Ю.Ю. Тарасевича. – Астрахань : Изд. дом «Астраханский университет», 2007. – 87 с.

7. Holland J. H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence [Text] / J. H. Holland. — The MIT Press, Cambridge, 1992. – 211 c.

8. Michalewicz Z. Genetic algorithms + Data Structures = Evolution Programs [Text] / Z. Michalewicz. – New York: Springer-Verlag, 1996. – 387 c.

9. Цой Ю.Р., Спицын В.Г. Генетический алгоритм / Спицын В.Г., Цой Ю.Р. Представление знаний в информационных системах : учебное пособие. – Томск : Изд-во ТПУ, 2006. – 146 с.

10. Mitchell M. An Introduction to Genetic Algorithms / M.Mitchell. – Cambridge: MIT Press, 1999. – 158 c.

11. Periaux J., Sefrioui M. Evolutionary computational methods for complex design in aerodynamics [Text] // AIAA-98-0222. – Reno, 1998. – 15 p.

12. Periaux J, Chen HQ, Mantel B, Sefrioui M, Sui HT (2001) Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. Finite Elem Anal Des 37(5), pp. 417–429.

13. Жирков А. Суперкомпьютеры: развитие, тенденции, применение. Обзор HPC-решений Eurotech [Текст] / А. Жирков // Современные методы автоматизации. – 2014. – № 2. – С. 16–20.

14. Arthur Trew (Editor), Greg Wilson (Editor). Past, Present, Parallel: A Survey of Available Parallel Computer Systems. – Springer, 1991. – 392 p.

Comparative Analysis of Scheduling Algorithms in Multiprocessor Systems

I.A. Molodkin, S.G. Svistunov
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russian Federation
imolodkin@gmail.com, ssg47@mail.ru

Abstract. **Descriptions of an algorithm of Degtyarev and the genetic algorithm applied to distribution of works in multiprocessor computing systems are provided. Results of the comparative analysis of quality of distribution of algorithms by comparison of the general operating time of system after distribution of works are given. The analysis is carried out by means of the developed program imitating model. It is shown that the Degtyaryov's algorithm provides fast result with good characteristics at short-term planning. It is simple in realization and works rather quicker genetic algorithm. Advantage of a genetic algorithm can be considered the best distribution of works at their small quantity and sufficient time for calculations. Recommendations about application of algorithms are formulated under different conditions.**

Keywords: **multiprocessor systems, algorithms, genetic algorithms, Degtyarev's algorithm.**

REFERENCES

1. Supercomputers. World market. – http://www.tadviser.ru (Accessed at 25.05.2018).

2. Multiprosessor Computers and Methods of their Projecting / B.A.Babayan, A.V.Bocharov, B.S.Volodin and others. – Moscow. : High School, 1990. – 384 p.

3. Lazarev A.A., Grafov E.R. Scheduling Theory. Tasks and Algorithms Moscow. – Moscow State University, 2011. – 222 p.

4. Degtyarev Yury Ivanovich. – https://mai.ru (Accessed at 25.05.2018). Available at: https://mai.ru/content/people/index.php?ID=28066.

5. Degtyarev Yu.I. Operations Research. – Moscow : High School, 1986. – 320 p.

6. Panchenko T.V. Genetic Algorithms [Text]: teaching aid / Yu.Yu. Tarasevich. Astrakhan: «Astrakhanskiy Universitet», 2007. – 87 p.

7. Holland J.H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence / J. H. Holland. The MIT Press, Cambridge, 1992. – 211 p.

8. Michalewicz Z. Genetic algorithms + Data Structures = Evolution Programs / Z. Michalewicz. – New York : Springer-Verlag, 1996. – 387 c.

9. Tsoy Yu.R., Spitsyn V.G. Genetic Algorithm / Spitsyn V.G., Tsoy Yu.R. Knowledge Representation in Information Systems: a Tutorial. – Tomsk : TPU, 2006. – 146 p.

10.Mitchell M. An Introduction to Genetic Algorithms [Текст]/M.Mitchell. – Cambridge: MIT Press, 1999. –158 p.

11.Periaux J., Sefrioui M. Evolutionary computational methods for complex design in aerodynamics // AIAA-98-0222. – Reno, 1998. – 15 p.

12. Periaux J, Chen HQ, Mantel B, Sefrioui M, Sui HT (2001) Combining game theory and genetic algorithms with application to DDM-nozzle optimization problems. Finite Elem Anal Des 37(5), pp. 417–429.

13. Zhirkov A Supercomputers: developing, trends, usage. Eurotech HPC solutions review [Superkomp'yutery: razvitie, tendentsii, primenenie. Obzor HPC-resheniy Eurotech] *Present automatization solutions* 2014, no. 2, p. 20

14. Arthur Trew (Editor), Greg Wilson (Editor). Past, Present, Parallel: A Survey of Available Parallel Computer Systems. – Springer, 1991. – 392 p.

Гомоморфное шифрование в базах данных

А.М. Щелкунов, М.Л. Глухарев
Петербургский государственный университет
путей сообщения Императора Александра 1
Санкт-Петербург, Россия
a_shchelkunov@mail.ru, mlgluharev@yandex.ru

***Аннотация*. Гомоморфное шифрование представляет собой криптографический примитив, который имеет большое прикладное значение, а также интересен с математической точки зрения. Несмотря на многолетние исследования в этой области, основные проблемы остаются нерешенными. В статье описываются основные особенности организации защищенных вычислений криптографическим примитивом на основе гомоморфного шифрования. Обсуждается проведение защищенных вычислений над данными клиента, хранящимися на удаленном недоверенном сервере в базе данных. Рассматривается пример защищенного сервера с облачной базой данных. Приводится описание экспериментального приложения на языке программирования Python для оценки производительности и пригодности алгоритмов, реализующих гомоморфную криптосистему с возможностью сложения и умножения зашифрованных данных в кольце Z_n. Обсуждаются результаты численных экспериментов с параметрами: уровень защищенности, типы выполняемых операций, максимальная длина операндов и время работы приложения.**

***Ключевые слова:* гомоморфная криптография, гомоморфное шифрование, матричные полиномы, криптографические вычисления, защита баз данных, защищенные вычисления.**

Введение

Одна из наиболее важных задач, стоящих перед современной криптографией, – проведение вычислений над зашифрованными данными без их расшифровки. Вопрос о возможности таких вычислений долгое время оставался открытым, и авторы схемы шифрования RSA полагали, что такие вычисления невозможны. Раздел криптографии, посвящённый схемам, допускающим вычисления над шифротекстами (шифртекстами), принято называть гомоморфной криптографией, а соответствующие схемы – гомоморфными. Различают полностью гомоморфные и частично гомоморфные схемы шифрования. В полностью гомоморфной схеме шифрования операции сложения и умножения шифротекстов являются гомоморфными, т. е. выполняются следующие соотношения:

$$D(E(m_1)\cdot E(m_2)) = m_1 \cdot m_2; \qquad (1)$$

$$D\left(E\left(m_1\right) + E\left(m_2\right)\right) = m_1 + m_2, \qquad (2)$$

где $E(\cdot)$ – функция шифрования, а $D(\cdot)$ – функция дешифрования.

Если в некоторой схеме шифрования выполняется одно (любое) из двух условий, то эта схема называется частично гомоморфной. Примеров частично гомоморфных шифровальных схем очень много. Например, схема RSA является гомоморфной относительно операции умножения, как и схема Эль-Гамаля. Схема RSA и схема Эль-Гамаля являются частично гомоморфными, а именно гомоморфными относительно операции умножения шифротекстов. Описания и разъяснения этих схем можно посмотреть в [1; 2].

В данной статье рассматриваются некоторые модели гомоморфных схем шифрования, полезные с практической точки зрения. Одной из интересных и практически ценных схем шифрования является предложенная в [3] схема, основанная на матричных полиномах. Преимущество этой модели в том, что все вычисления, проводимые над зашифрованными данными, сводятся к сложению и перемножению матриц, а эти операции, как известно, допускают широкое распараллеливание. Возможность распараллеливания вычислений повышает практическую значимость схемы шифрования.

Защищенные вычисления

Одним из самых естественных приложений гомоморфного шифрования является проведение защищенных вычислений над данными клиента, хранящимися на удаленном недоверенном сервере в базе данных. Поясним более подробно, что под этим подразумевается.

Допустим, клиент зашифровал данные стандартным шифром и разместил их на удалённом сервере. В случае необходимости их изменения клиент может доверить серверу свой секретный ключ, для того чтобы сервер расшифровал данные и внёс нужные клиенту изменения, а затем зашифровал данные снова. Но в этом случае сервер сможет прочитать все данные клиента. Также возможен перехват ключа при передаче его серверу по каналу, в результате чего данные клиента могут стать доступными некоему третьему лицу. Также клиент может скачать свои данные из сервера на свой компьютер, расшифровать, провести над ними нужное вычисление, при необходимости зашифровать этот результат и загрузить его на сервер в базу данных. Однако это потребует много времени и вычислительных ресурсов, которыми клиент, возможно, не располагает.

Из сказанного выше вытекают несколько конкретных требований к безопасному (защищенному) серверу. Во-первых, данные клиента должны храниться в таком виде, чтобы при их чтении невозможно было понять, что это за данные (данные необходимо шифровать). Причем данные должны поступать на сервер уже зашифрованными. Следовательно, шифрование должно проводиться на стороне клиента. Во-вторых, должна быть возможность обрабатывать эти данные, не расшифровывая. Иначе сервер с базой данных становится всего лишь безопасным хранилищем. А для каждой операции над данными потребуется пересылать их на сторону клиента.

В настоящее время существующие базы данных не являются полностью защищенными. В лучшем случае есть возможность лишь зашифровать данные на стороне пользователя, но это бывает крайне редко. Довольно часто данные зашифрованы ключом, который хранится на сервере вместе с базой данных.

Перейдем к рассмотрению конкретного примера защищенного сервера с базой данных. Речь пойдет об организации защищенной облачной базы данных. Задача будет следующей: Есть организация, которая обладает достаточными ресурсами для хранения и обработки своей базы данных. Серверу организация не доверяет. Поэтому все данные шифруются перед отправкой на сервер. Клиенты, обращающиеся к базе данных, являются программами, запущенными на машинах, находящихся во владении организации. Все клиенты доверяют друг другу, и ключ расшифрования у них является общим. Далее рассмотрим как эту задачу возможно решить с помощью симметричного гомоморфного шифрования.

Модель базы данных

Современная реляционная база данных является набором прямоугольных таблиц. Предположим, что таблица только одна. Таблица имеет m атрибутов a_1, ..., a_m и по сути представляет из себя набор записей $\{R_i\}_{i=1}^{m}$, где $R_i = \{\upsilon_{i,j}\}_{j=1}^{m}$, $\upsilon_{i,j}$ – значение записи R_i для атрибута a_j. Также еще договоримся, что a_j, $\upsilon_{i,j}$ и номера записей $i \in \{1, ..., w\}$ являются элементами некоторого большого конечного поля F_q. Клиенту необходимо делать простые SQL-запросы к базе данных, такие как:

$$\text{SELECT * FROM db WHERE } (a_{z_1} = \upsilon_1^*) \text{ OR...OR } (a_{z_t} = \upsilon_t^*); \quad (3)$$

$$\text{SELECT * FROM db WHERE } (a_{z_1} = \upsilon_1^*) \text{ AND...AND } (a_{z_t} = \upsilon_t^*). \quad (4)$$

Запросы типа (3) будем называть дизъюнктивными, а запросы типа (4) – конъюнктивными.

Обратимся теперь к модели *клиент–сервер*. Наш сервер S хранит $db = \{R_i\}_{i=1}^{w}$, принадлежащую клиенту K. Периодически наш клиент K обращается с запросами к серверу S. В результате запросов клиент K должен получить все записи в db, удовлетворяющие условию WHERE. При этом S ничего не должен узнать о значениях υ_i^*, $1 \le i \le t$, участвующих в запросе, а также о том, какие записи в db соответствуют запросу.

Данный подход к решению этой задачи заключается в том, что обработка запроса проходит в две стадии.

Сначала K получает номера i_1, ..., $i_n \in \{1, ...,w\}$ записей, который соответствуют условиям в SQL запросе. Этот этап нужно организовать так, чтобы S не узнал υ_i^*,

$$1 \le i \le t \text{ и } i_1, ..., i_n.$$

Клиент K по очереди извлекает из db записи с индексами i_1, ..., i_n. При этом S не должен узнать i_1, ..., i_n.

Эта последовательность действий использовалась, например, в [4]. Отметим, что в [4] рассматривалась немного другая задача. Предполагалось, что база данных принадлежит S и, как следствие, она хранилась на S в открытом виде. Мы же рассматриваем случай, когда база данных принадлежит клиенту K. Поэтому S хранит не $db = \{\upsilon_{i,j}\}_{i=1,j=1}^{w,m}$, а $db = \{E_{sk}(\upsilon_{i,j})\}_{i=1,j=1}^{w,m}$, где E – отображение шифрования, sk – секретный ключ, принадлежащий клиенту K.

Поочередно рассмотрим, как два этапа обработки запроса можно организовать при условии, что для шифрования $\upsilon_{i,j}$ используется криптосистема с гомоморфными свойствами.

Защищенное получение индексов

Поскольку клиент K хочет скрыть υ_i^*, $1 \le i \le t$, ему необходимо их зашифровать. На сервер S он передает пары $(a_{Z_k}, E(\upsilon_k^*))$, $1 \le i \le t$. В свою очередь S должен для каждой записи $R_i = \{E(\upsilon_{i,j})\}_{j=1}^{m}$, i = 1..w, провести следующее вычисление:

для $\forall_{z_k}$ = 1…t сервер S должен вычислить $e_k = f(E(\upsilon_{i,z_k}, E(\upsilon_k^*))$, где f – функция, осуществляющая проверку на равенство υ_{i,z_k}, и υ_k^* гомоморфно. То есть $e_k = E(0)$, если $\upsilon_{i,z_k} \ne \upsilon_k^*$, и $e_k = E(1)$, если $\upsilon_{i,z_k} = \upsilon_k^*$.

Далее в зависимости от типа запроса S должен сделать следующее:

– конъюнктивный запрос: нужно вычислить

$$e_i^{'} = Hom_{AND}(e_1...e_t), \quad (3)$$

где Hom_{AND} – функция, осуществляющая вычисление побитовой конъюнкции нижележащих открытых текстов. Ясно, что если $e_i^{'} = E(1)$, то Ri соответствует запросу, а если $e_i^{'} = E(0)$, то не соответствует. Отметим, что выполняется $Hom_{AND}(e_1,...,e_t) = e_1 \cdot ... \cdot e_t$;

– дизъюнктиный запрос: нужно вычислить

$$e_i^{'} = Hom_{XOR}(e_1...e_t), \quad (4)$$

где Hom_{XOR} – функция, вычисляющая побитовый XOR нижележащих открытых текстов.

Сервер S отправляет вектор шифртекстов $\vec{\mathrm{Re}}\, s = (e_1^{'},...,e_w^{'})$ клиенту K.

Расшифровав компоненты вектора $\vec{\mathrm{Re}}\, s$, клиент K узнает индексы i_1, ..., i_n записей, соответствующих его SQL-запросу. Отметим, что в общем представленное решение пока не очень эффективно, так как для того, чтобы получить индексы, K должен обработать объем информации в размере по сути одного столбца таблицы db. Вопрос о том, как его сократить, требует дальнейшей проработки.

Ясно, что для реализации описанного подхода можно использовать симметричную гомоморфную криптосистему.

Для извлечения записей из базы данных по их индексам наш клиент K может воспользоваться стандартным протоколом секретного получения информации [5; 6].

Реализация гомоморфного шифрования

В качестве практического этапа исследования разработано экспериментальное приложение на языке Python для оценки производительности и пригодности алгоритмов, реализующее полностью гомоморфную криптосистему с возможностью сложения и умножения над зашифрованными данными в кольце Zn.

Алгоритм, реализованный в работе [7], предполагает расширение кольца Z_2 на кольцо Zn, где n – некоторое достаточно большое натуральное число. Возможности языка Python позволяют задавать значение n произвольно большим, практически ограничивая его значение лишь размером доступной оперативной памяти машины.

Данное требование, связанное с вычислениями в кольце Zn, означает, что операнды вычислений должны быть также неотрицательными числами, не превосходящими n.

Кроме того, для корректности вычислений, сумма и произведение двух чисел не должны превосходить значение n. Это требование кажется сложно выполняемым, так как заранее нельзя ограничивать размер чисел, получаемых в результате умножения произвольных операндов. Но у нас есть возможность выбрать модуль n заведомо достаточно большим.

Алгоритм безопасных вычислений

Алгоритм с использованием гомоморфного шифрования может быть описан следующим образом:

1) генерация публичного и приватного ключей;
2) шифрование операндов;
3) применение к операндам одной из допустимых функций (сложение или умножение);
4) дешифрование полученного на предыдущем шаге результата.

Рассмотрим этот процесс более детально. В качестве открытого ключа будем использовать некоторое число p, взаимно простое с числом n и с порядком группы Zn. Таким образом, выбираем p такое, что наибольший общий делитель p и порядка группы равнялся единице.

Процесс шифрования определяется соотношением вида:

$$c = z + pq = m + n * r + pq. \quad (5)$$

Для процесса дешифрования соотношение имеет вид:

$$m = (c \bmod p) \bmod n. \quad (6)$$

Процесс шифрования требует от пользователя еще двух дополнительных значений: r и q. Отталкиваясь от этой потребности в качестве публичного ключа, будем использовать массив *PublicKey* из 100 элементов, где i-й элемент задан следующим способом:

$$PublicKey[i] = p * randomValue + n * r. \quad (7)$$

Здесь randomValue – случайное значение, n – порядок группы, r – случайное число в диапазоне (10, 100).

Видоизменим формулу шифрования следующим образом:

$$c = m + \sum PublicKey *. \quad (8)$$

Здесь $\sum$ *PublicKey** – сумма некоторого случайного набора приватных ключей, разный для каждого из операндов. Случайность данных наборов обусловливает различие констант шифрования.

Нетрудно проверить, что данный выбор удовлетворяет приведенным выше формулам для шифрования и дешифрования.

На рисунке 1 представлен пример шифрования двух целых чисел и выполнения операций сложения над ними. Данные отображаются в hex-формате и в виде символов таблицы ASCII.

Для исследования эффективности реализации системы гомоморфного шифрования был проведен эксперимент, параметрами которого являлись уровень защищенности, типы выполняемых операций, максимальная длина операндов и время работы. Уровень защищенности – характеристика, определяющая значение модуля группы и длину используемых ключей шифрования. В эксперименте использовались модуль и ключи, длина которых составляла от 5 до 8000 десятичных цифр. Используемые операции – шифрование и дешифрование операндов, умножение и сложение операндов. Данные получены для двух значений максимальной длины операндов – 1e10 и 1e20, т. е. для 10-значных и 20-значных десятичных чисел.

Оценки получены при 1000-кратном повторении вычислений с заданными параметрами на компьютере класса Intel(R) CORE (TM) i3-2728 CPU @ 2.20 GHz RAM 4 GB [6].

Результаты эксперимента приведены в таблице и на рисунке 2.

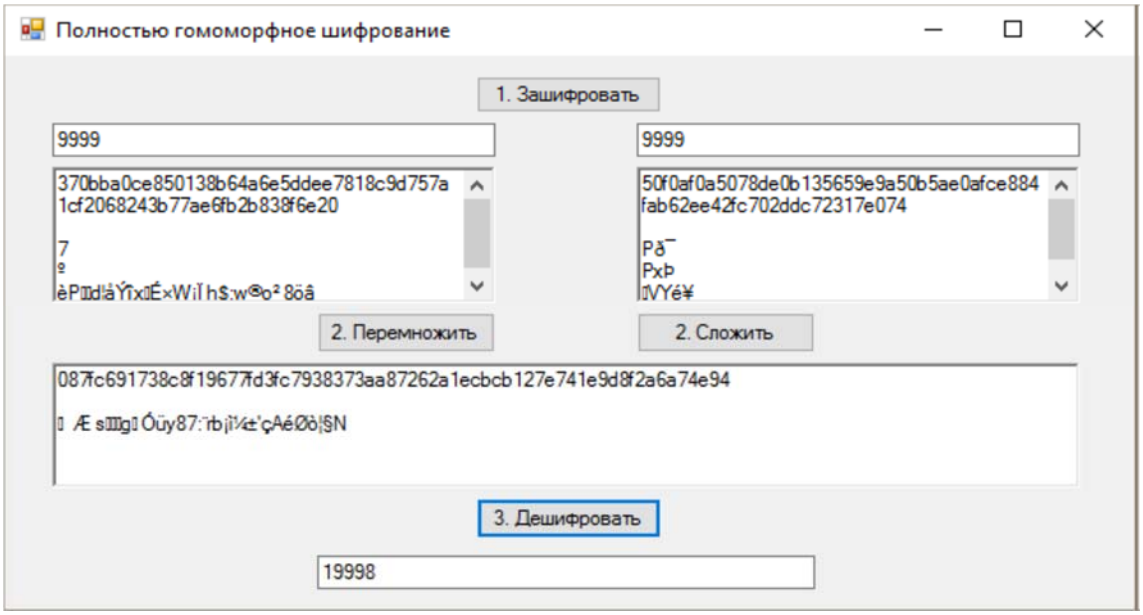


Рис. 1. Пример визуализации работы алгоритма безопасных вычислений

ТАБЛИЦА. Характеристики реализации гомоморфного шифрования

Уровень защищенности (макс. кол-во цифр ключа)	Макс. длина операндов = 1010		Макс. длина операндов = 1020	
	Шифрование, дешифрование, сложение, с	Шифрование, дешифрование, умножение, с	Шифрование, дешифрование, сложение, с	Шифрование, дешифрование, умножение, с
5	0,92	0,85	1,07	0,86
10	0,93	0,84	1,04	1,09
20	0,93	0,84	1,03	0,93
50	0,94	0,85	1,04	0,94
100	0,94	0,86	1,04	0,95
200	0,94	0,90	1,04	0,99
500	1,00	1,06	1,11	1,18
1000	1,14	1,52	1,27	1,66
2000	1,59	3,12	1,76	3,68
4000	3,21	8,90	3,64	10,68
8000	9,89	31,57	11,31	35,38

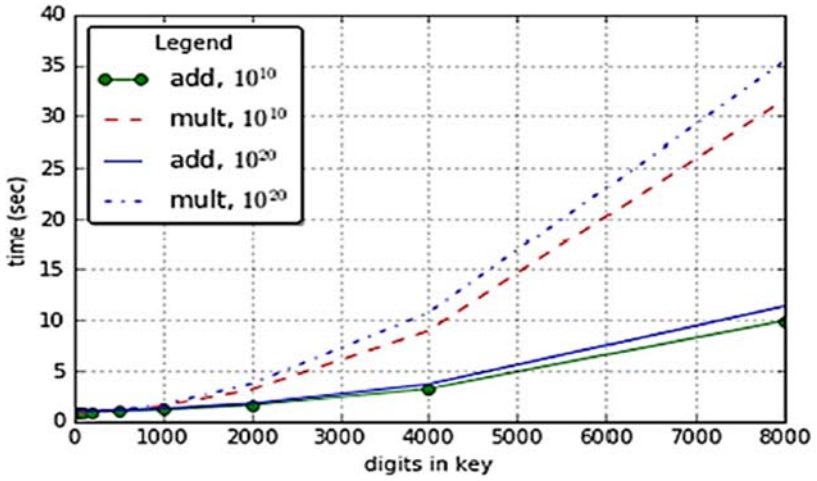


Рис. 2. Зависимости времени реализации алгоритма от параметров шифрования

Проведенный эксперимент подтвердил эффективность использования рассматриваемого подхода к организации безопасных вычислений. Выбор достаточно большого по сравнению со значениями используемых операндов модуля позволяет корректно решать проблему однозначности вычислений. Операции вычитания и деления на целое число, наличие которых необходимо для организации полноценных вычислений, могут быть реализованы через операции сложения и умножения на обратное число (в этом случае модуль группы должен быть простым числом).

Заключение

К сожалению, в настоящее время авторам не известна ни одна реализация гомоморфного шифрования, полностью готовая к внедрению в реальные системы.

Для того чтобы гомоморфное шифрование было эффективно применимо, должна быть реализация, удовлетворяющая как минимум следующим требованиям:

1. Поддерживаемые математические функции должны покрывать нужды программистов.

2. Диапазоны значений чисел должны покрывать по крайней мере стандартные типы данных, а вычисления, производимые над зашифрованными данными, соответствующие такому размеру чисел, – иметь приемлемую производительность.

3. Точность и скорость вычислений не должны уменьшаться или ухудшаться во время вычислений.

4. Количество разнообразных ключей должно быть достаточным, чтобы исключить атаку полным перебором.

Тем не менее гомоморфное шифрование является одним из самых мощных математических аппаратов для сохранности данных в различных прикладных средах. Но только полностью гомоморфное шифрование способно исключить необходимость хотя бы частичной расшифровки данных для произведения вычислений над ними. Впрочем и оно не способно вытеснить любые другие виды криптографической защиты, поскольку любое подобное шифрование принципиально уязвимо для атаки с подобранным текстом. Для примера отметим, что современные результаты, касающиеся способов шифрования с открытыми ключами и блочного шифрования, рассматриваются в статьях [10; 11].

Литература

1. Rivest R. L, Adleman L., Dertouzos M. L. On data banks and privacy homomorphisms // Foundations of secure computation, 1978. Vol. 32, no. 4, p. 169–178.

2. ElGamal encryption [Elektronnyy resurs] / Vikipediya. Rezhim dostupa: https://en.wikipedia.org/wiki/ ElGamal_ encryption. Data dostupa: 29.05.2018.

3. Буртыка Ф. Б. Симметричное полностью гомоморфное шифрование с использованием неприводимых матричных полиномов // Изв. ЮФУ. Техн. науки. – 2014. – С. 107–122.

4. Song D., Wagner D., Perrig A. Practical techniques for searches on encrypted data // SP ’00 Proc. 2000 IEEE Symp. Security and Privacy. – Berkeley: Univ. California, 2000.

5. Boneh D., Gentry C., Halevi S., Wang D., Wu D.J. Private database queries using somewhat homomorphic encryption // Applied cryptography and network security Springer, 2013, p. 102–118.

6. Boneh D., Gentry C., Halevi S., Wang F., Wu D.J. Private database queries using somewhat homomorphic encryption // Applied Cryptography and Network Security: 11th Int. Conf., ACNS 2013, Banff, AB, Canada, June 25–28, 2013. Proc. / M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, eds. – Berlin: Springer, 2013. – (Lect. Notes Comp. Sci. Security Cryptology; Vol. 7954). – P. 102–118.

7. Gasarch W. A survey on private information retrieval // Bulletin of the eatcs Citeseer, 2004.

8. Варновский Н.П. Гомоморфное шифрование / Н.П. Варновский, А. В. Шокуров // Труды Института ситемного программирования. Т. 12 ; под ред. В. П. Иванникова. – М. : ИСП РАН, 2006. – С. 27–36.

9. Gentry C. A Fully Homomorphic Encryption Scheme: Ph. D. thesis. – Stanford Univ., 2009.

10. Молдовян Н.А., Вайчикаускас М.А. Генерация степенных сравнений как способ открытого шифрования и протокол отрицаемого шифрования // Интеллектуальные технологии на транспорте. 2018. № 1(13). – С. 32–37. – http://www.itt-pgups.ru.

11. Молдовян А.А., Татчина Я.А. Способы псевдовероятностного блочного шифрования // Интеллектуальные технологии на транспорте. – 2018. № 1(13). – С. 25–31. – http://www.itt-pgups.ru.

Homomorphic Encryption in Databases

A.M. Shcelkunov, M.L. Glukharev
Emperor Alexander I St. Petersburg State
Transport University
St. Petersburg, Russia
a_shchelkunov@mail.ru, mlgluharev@yandex.ru

Abstract. **Homomorphic encryption is a cryptographic primitive. It has a large application value, and is also interesting from a mathematical point of view. Despite years of research in this field, the main problems remain unresolved. The article describes the main features of the organization of protected computing by a cryptographic primitive based on homomorphic encryption. It is discussed how to perform secure calculations on client data stored on a remote, untrusted server in the database. An example of a secure server with a cloud database is considered. The description of the experimental application in the programming language Python for evaluating the performance and suitability of algorithms that implement a homomorphic cryptosystem with the possibility of addition and multiplication over encrypted data in the Zn ring is described. The results of numerical experiments with parameters are discussed: the level of security, the types of performed operations, the maximum length of the operands, and the running time of the application.**

Keywords: **homomorphic cryptography, homomorphic encryption, matrix polynomials, cryptographic calculations, data protection, secure computing, computer security.**

REFERENCES

1. Rivest R. L, Adleman L., Dertouzos M. L. On data banks and privacy homomorphisms // Foundations of secure computation, 1978. Vol. 32, no. 4, p. 169–178.
2. ElGamal encryption [Elektronnyy resurs] / Vikipediya. Rezhim dostupa: https://en.wikipedia.org/ wiki/ElGamal_ encryption. Data dostupa: 01.04.2016.
3. Burtyka F. B. Simmetrichnoe polnost'yu gomomorfnoe shifrovanie s ispol'zovaniem neprivodimykh matrich-nykh polinomov // Izv. YuFU. Tekhn. nauki. – 2014. p. 107–122.
4. Song D., Wagner D., Perrig A. Practical techniques for searches on encrypted data // SP ’00 Proc. 2000 IEEE Symp. Security and Privacy.–Berkeley: Univ. California, 2000.
5. Boneh D., Gentry C., Halevi S., Wang D., Wu D. J. Private database queries using somewhat homomorphic encryption // Applied cryptography and network security Springer, 2013, p. 102–118.
6. Boneh D., Gentry C., Halevi S., Wang F., Wu D. J. Private database queries using somewhat homomorphic encryption // Applied Cryptography and Network Security: 11th Int. Conf., ACNS 2013, Banff, AB, Canada, June 25–28, 2013. Proc. / M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, eds. – Berlin : Springer, 2013. – (Lect. Notes Comp. Sci. Security Cryptology; Vol. 7954). – P. 102–118.
7. Gasarch W.. A survey on private information retrieval // Bulletin of the eatcs Citeseer, 2004.
8. Varnovskiy, N.P. Gomomorfnoe shifrovanie / N.P. Varnovskiy, A.V. Shokurov // Trudy Instituta Sitemnogo programmirovaniya: Tom 12. (pod red. V. P. Ivannikova). – M. : ISP RAN, 2006, c. 27–36.
9. Gentry C. A Fully Homomorphic Encryption Scheme: Ph. D. thesis.–Stanford Univ., 2009.
10. Moldovyan N. A., Vaychikauskas M. A. Generatsiya stepennykh sravneniy kak sposob otkrytogo shifrovaniya i protokol otritsaemogo shifrovaniya // Intellek-tual'nye tekhnologii na transporte. – 2018. – No 1(13). – P. 32–37. – http://www.itt-pgups.ru.
11. Moldovyan A. A., Tatchina Ya. A. Sposoby psevdoveroyatnostnogo blochnogo shifrovaniya // Intellektual'nye tekhnologii na transporte. – 2018. – No 1(13). – P. 25–31. – http://www.itt-pgups.ru.

Выбор оптимального метода минимизации при разработке программы поиска минимальной дизъюнктивной нормальной формы

В.А. Табанина
Петербургский государственный университет
путей сообщения Императора Александра I
Санкт-Петербург, Россия
alekseevnav@gmail.com

***Аннотация*. Рассмотрены распространенные в изучении методы минимизации логических функций: эквивалентных преобразований, карт Карно, Куайна и Куайна–Маккласки. Проводится их анализ и сравнение, указываются основные математические соотношения, лежащие в основе методов, достоинства и недостатки. Выбран оптимальный метод с целью разработки Java-приложения, реализующего минимизацию логических функций в случае произвольного числа переменных.**

***Ключевые слова:* минимизация, дизъюнктивные нормальные формы, совершенная и минимальная дизъюнктивные нормальные формы, эквивалентные преобразования, карты Карно, Куайн, Маккласки, логические функции.**

Введение

В настоящее время вычислительная техника становится неотъемлемой частью нашей жизни. Каждый год появляются новые сферы ее применения. Из-за этого задачи, выполняемые вычислительными системами, усложняются. Для их решения требуются разработки новых и совершенствование старых методов логического проектирования цифровых устройств. Логическое проектирование при этом понимается в широком смысле. Оно основывается на методах минимизации логических функций и включает в себя не только структуру функции, но и анализ динамики на уровне переходных процессов, который связан с изменением переменных и временными характеристиками элементов. В связи с этим исследования, направленные на развитие методов минимизации логических функций, по-прежнему актуальны.

Практически любая логическая функция может быть упрощена непосредственно с помощью законов алгебры логики, но, как правило, при таком подходе преобразования требуют громоздких вычислений, тратится много времени и вероятность допущения ошибки возрастает. Поэтому целесообразно использовать специально разработанные алгоритмические методы минимизации, позволяющие проводить упрощение функции быстрее и безошибочно.

В учебной дисциплине «Алгебра логики» чаще всего рассматриваются следующие методы:

- метод эквивалентных преобразований;
- карты Карно;
- метод Куайна;
- метод Куайна–Маккласки;

Целью настоящей статьи является сравнение и анализ этих методов, определение оптимального для минимизации логических функций произвольного числа переменных, а также разработка компьютерной программы для поиска минимальных дизъюнктивных форм выбранным методом на языке Java.

Основные термины

Функцией алгебры логики n-переменных (или функцией Буля) называется функция n-переменных, где каждая переменная принимает два значения: 0 или 1, при этом функция может принимать только одно из двух значений.

Дизъюнктивной нормальной формой (ДНФ) формулы A называется равносильная ей формула, представляющая собой дизъюнкцию элементарных конъюнкций.

Минимальная дизъюнктивная нормальная форма получается из совершенной дизъюнктивной нормальной формы путем применения указанных ранее методов.

Элементарной конъюнкцией n-переменных называется конъюнкция переменных или их отрицаний.

Среди множества ДНФ A существует единственная ДНФ A', из которой имеются перечисленные ниже свойства [1]:

1) каждое логическое слагаемое формулы содержит все переменные, входящие в функцию $f(X_1,X_2,\ldots, X_n)$;
2) все логические слагаемые формулы различны;
3) ни одно логическое слагаемое формулы не содержит одновременно переменную и ее отрицание;
4) ни одно логическое слагаемое формулы не содержит одну и ту же переменную дважды.

Эти свойства называются свойствами совершенства. ДНФ, удовлетворяющая этим свойствам, называется совершенной дизъюнктивной нормальной формой (СДНФ).

Если функция $f(X_1,X_2,\ldots, X_n)$ задана таблицей истинности, то соответствующая ей СДНФ может быть получена просто. А именно, для каждого набора значений переменных, на котором функция $f(X_1,X_2,\ldots,X_n)$ принимает значение 1, записывается конъюнкция элементарных переменных, взяв за член конъюнкции $X_к$, если значение $X_к$ на указанном наборе значений переменных есть 1, и отрицание $X_к$, если значение $X_к$ есть 0.

Дизъюнкция всех записанных конъюнкций и будет искомой формулой. Такие наборы конъюнкций называются импликантами. Элементарная конъюнкция K называется

импликантой функции f, если для всякого набора $a = (a_1,\ldots, a_n)$ из 0 и 1 условие $K(a) = 1$ влечет $f(a) = 1$.

Задача минимизации ДНФ заключается в поиске такой ДНФ для заданной булевой функции f, которая содержала бы минимальное число конъюнкций или букв [2]. ДНФ называется минимальной, если она имеет наименьшую длину среди всех эквивалентных ей ДНФ. Дизъюнкция простых импликант булевой функции называется безызбыточной (тупиковой), если она представляет эту функцию и не содержит импликант, поглощаемых другими [3].

Из тупиковой ДНФ нельзя удалить ни одной элементарной конъюнкции и ни одной буквы без изменения представляемой ею булевой функции. Тупиковая ДНФ получается из сокращенной ДНФ путем последовательного исключения простых импликант. Для одной и той же функции может существовать (в общем случае) несколько различных тупиковых ДНФ, в отличие от сокращенной ДНФ, которая однозначно определена. Справедливы следующие утверждения [4]:

1. Любая МДНФ булевой функции представляет собой дизъюнкцию множества некоторых простых импликант этой функции и являются тупиковой ДНФ.
2. Булева функция может иметь несколько различных минимальных ДНФ.
3. Существуют такие тупиковые ДНФ, которые не являются минимальными.

Характеристика методов минимизации

Рассмотрим основные характеристики методов минимизации, ключевые математические соотношения, а также достоинства и недостатки методов.

Метод эквивалентных преобразований

В основе метода минимизации булевых функций с помощью эквивалентных преобразований лежит последовательное использование законов булевой алгебры [5].

Основные эквивалентные соотношения в булевой алгебре

1. Ассоциативность конъюнкции и дизъюнкции:
$$X_1 \wedge X_2 = X_2 \wedge X_1 \text{ и } X_1 \vee X_2 = X_2 \vee X_1.$$
2. Коммуникативность конъюнкции и дизъюнкции:
$$X_1 \wedge (X_2 \wedge X_3) = (X_1 \wedge X_2) \wedge X_3 = X_1 \wedge X_2 \wedge X_3;$$
$$X_1 \vee (X_2 \vee X_3) = (X_1 \vee X_2) \vee X_3 = X_1 \vee X_2 \vee X_3.$$
3. Дистрибутивность конъюнкции относительно дизъюнкции:
$$X_1 \wedge (X_2 \vee X_3) = X_1 \wedge X_2 \vee X_1 \wedge X_3.$$
4. Дистрибутивность дизъюнкции относительно конъюнкции:
$$X_1 \vee (X_2 \wedge X_3) = (X_1 \vee X_2) \wedge (X_1 \vee X_3).$$
5. Идемпотентность:
$$X_1 \wedge X_1 = X_1 \text{ и } X_1 \vee X_1 = X_1.$$
6. Закон двойного отрицания:
$$\overline{\overline{X}} = X.$$
7. Свойства констант 0 и 1:
$$X \wedge 1 = X; \quad X \wedge 0 = 0;$$
$$X \vee 1 = 1; \quad X \vee 0 = X.$$
8. Правила Де Моргана:
$$\overline{X_1 \wedge X_2} = \overline{X_1} \vee \overline{X_2};$$
$$\overline{X_1 \vee X_2} = \overline{X_1} \wedge \overline{X_2}.$$
9. Закон противоречия:
$$X \wedge \overline{X} = 0.$$
10. Закон исключенного третьего:
$$X \vee \overline{X} = 1.$$

Эти эквивалентные соотношения выводимы друг из друга, и их достаточно для выполнения всех эквивалентных преобразований. Однако для упрощения логических формул используются следующие соотношения, полученные с помощью основных эквивалентных преобразований:

1) склеивание
$$AX \vee A\overline{X} = A(X \vee \overline{X}) = A;$$
2) поглощение
$$AX \vee A = A,$$

где A – любая логическая функция.

Этот метод требует больших затрат труда, поэтому его стоит использовать лишь для простых функций и для количества логических переменных не более четрех. В случае, когда логических переменных много и функция сложная, возрастает вероятность пропустить склеивающиеся импликанты, что приведет к неправильной минимизации логической функции. Кроме того, в связи с неалгоритмичностью его сложно запрограммировать.

Минимизация с помощью карт Карно

Минимизация с помощью карт Карно – графический метод представления таблиц истинности логической функции. На карте Карно булева функция $f(X_1, X_2,\ldots, X_n)$ задается указанием в каждой ячейке значения, которое она принимает на соответствующем наборе простых элементов [6]. Количество ячеек зависит от числа логических переменных, а именно: таблица содержит 2^n ячеек, где n – число переменных.

Все эти наборы переменных (импликанты) образуют структуру, по топологии эквивалентную N-мерному кубу. Причем любые две импликанты, соединенные ребром, могут быть склеены и поглощены. На рис. 1 в качестве примера показана таблица истинности для булевой функции трёх переменных в виде трёхмерного куба и соответствующая ему развертка.

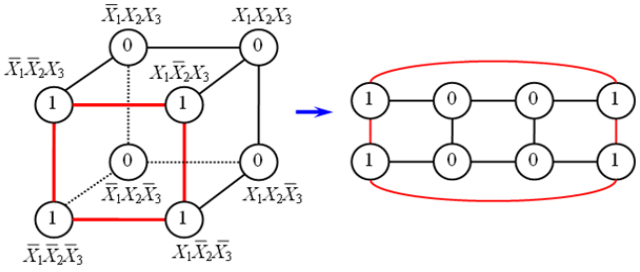


Рис. 1. Таблица истинности для булевой функции трёх переменных и соответствующий ей куб

Этот метод применим при трех, четырех и даже пяти переменных. Он основан на зрительном анализе, поэтому при большом количестве переменных таблицы становятся громоздкими и менее понятными, что может привести к допущению ошибок. Данный метод с трудом может быть применен для реализации вычислительной техникой.

Метод Куайна

Метод Куайна условно разделен на два этапа [7]:

- преобразование совершенной дизъюнктивной нормальной формы в сокращенную;
- преобразование сокращенной формы в минимальную.

Первый этап. В заданной СДНФ все импликанты нумеруются для удобства дальнейшего склеивания. После производится попарное сравнение членов СДНФ, их анализ на возможность склеивания и поглощения. Полученный набор переменных называют простыми импликантами [8].

Второй этап. Сокращенная дизъюнктивная нормальная форма может содержать в себе так называемые лишние простые импликанты. Лишними импликантами называют такие простые импликанты, при удалении которых результат никак не изменится. Чтобы выявить такие импликанты, нужно составить импликантную матрицу Куайна (рис. 2), в строки которой записываются полученные на прошлом этапе простые импликанты, а в столбцы – члены СДНФ [9]. После заполнения таблицы рассматриваются все пересечения строк со столбцами и ищется полное покрытие таблицы с минимальным количеством простых импликант.

Метод Куайна–Маккласки

В методе Куайна есть один существенный недостаток, который связан с попарным сравнением всех импликант. Это сравнение занимает значительное количество времени, а также есть большая вероятность пропустить какое-нибудь склеивание или поглощение, что приведет к ошибке. Эдвард Дж. Маккласки усовершенствовал метод Куайна, систематизировав первый этап.

Аналогично методу Куайна метод Куайна–Маккласки условно разделен на два этапа, и в его основу тоже положен закон склеивания и поглощения. В отличие от рассмотренного ранее метода Маккласки приводит все элементарные конъюнкции к двоичному номеру, т. е. заменяет X на 1, а $\overline{X}$ – на 0 [10]. После этого все двоичные номера сортируются по количеству единиц и записываются в таблицу со столбцами «номер группы», который соответствует количеству единиц в импликантах, и «двоичные номера» [11]. Склеивание производится только между соседними группами и между теми импликантами, в которых отличается только одна переменная.

Второй этап в методе Куайна–Маккласки практически не отличается от второго этапа в методе Куайна. Разница лишь в том, что в строки и столбцы импликантной таблицы записываются не члены СДНФ и простые импликанты, а их двоичные номера.

Усовершенствование МакКласки – разбиение конституент на группы по количеству в них единиц – уменьшило число попарных сравнений при склеивании, что значительно снизило риск сделать ошибку и сократило время на выполнение ручного счета, а также упростило программную реализацию для любого числа переменных.

Исходя из проведенного анализа, для реализации программы минимизации логических функций произвольного числа переменных взят метод Куайна–Маккласки, так как он алгоритмизирован и подходит для любого числа импликант.

Пример работы и интерфейс программы показан на рис. 3.

Конституенты единицы. Простые импликанты	$\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4\bar{X}_5\bar{X}_6$	$\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4\bar{X}_5X_6$	$\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_4X_5\bar{X}_6$	$\bar{X}_1\bar{X}_2\bar{X}_3X_4\bar{X}_5\bar{X}_6$	$\bar{X}_1\bar{X}_2\bar{X}_3X_4\bar{X}_5X_6$	$\bar{X}_1\bar{X}_2X_3\bar{X}_4\bar{X}_5\bar{X}_6$	$\bar{X}_1\bar{X}_2X_3\bar{X}_4\bar{X}_5X_6$	$\bar{X}_1\bar{X}_2X_3X_4\bar{X}_5\bar{X}_6$	$\bar{X}_1X_2\bar{X}_3\bar{X}_4\bar{X}_5\bar{X}_6$	$\bar{X}_1X_2\bar{X}_3X_4\bar{X}_5\bar{X}_6$	$\bar{X}_1X_2X_3\bar{X}_4\bar{X}_5\bar{X}_6$	$\bar{X}_1X_2X_3X_4\bar{X}_5\bar{X}_6$	$X_1\bar{X}_2\bar{X}_3\bar{X}_4\bar{X}_5\bar{X}_6$	$X_1\bar{X}_2\bar{X}_3X_4\bar{X}_5\bar{X}_6$	$X_1\bar{X}_2X_3\bar{X}_4\bar{X}_5\bar{X}_6$	$X_1X_2\bar{X}_3\bar{X}_4\bar{X}_5\bar{X}_6$	$X_1X_2\bar{X}_3X_4\bar{X}_5\bar{X}_6$	$X_1X_2X_3\bar{X}_4\bar{X}_5\bar{X}_6$	$X_1X_2\bar{X}_3\bar{X}_4X_5\bar{X}_6$	$X_1X_2X_3\bar{X}_4X_5\bar{X}_6$	$X_1X_2X_3X_4\bar{X}_5\bar{X}_6$	$X_1X_2X_3X_4X_5\bar{X}_6$	$X_1X_2X_3X_4X_5X_6$
$\bar{X}_1\bar{X}_2\bar{X}_3\bar{X}_6$	X	X	X		X																	
$\bar{X}_1\bar{X}_3\bar{X}_4X_5\bar{X}_6$		X						X														
$\bar{X}_1\bar{X}_2\bar{X}_3X_4\bar{X}_5$			X	X																		
$\bar{X}_1\bar{X}_2X_4\bar{X}_5\bar{X}_6$			X			X																
$\bar{X}_1\bar{X}_3X_4\bar{X}_5\bar{X}_6$			X						X													
$\bar{X}_2\bar{X}_3X_4\bar{X}_5\bar{X}_6$			X									X										
$X_2\bar{X}_3\bar{X}_4\bar{X}_5X_6$							X								X							
$\bar{X}_1X_2\bar{X}_4X_5\bar{X}_6$								X		X												
$X_2\bar{X}_3\bar{X}_4X_5\bar{X}_6$								X								X						
$X_1X_2\bar{X}_3\bar{X}_4\bar{X}_5$														X	X							
$X_1X_2\bar{X}_3\bar{X}_4\bar{X}_6$														X		X						
$X_1X_2\bar{X}_3\bar{X}_5\bar{X}_6$															X		X					
$X_1\bar{X}_2X_3X_4\bar{X}_5X_6$													X									
$X_1X_2X_3X_4\bar{X}_5\bar{X}_6$																			X			
$X_2X_3X_4X_5X_6$											X											X
$X_1X_2X_4X_5X_6$																		X				X
$X_1X_2\bar{X}_3X_4X_6$																	X	X				

Рис. 2. Пример заполненной матрицы Куайна

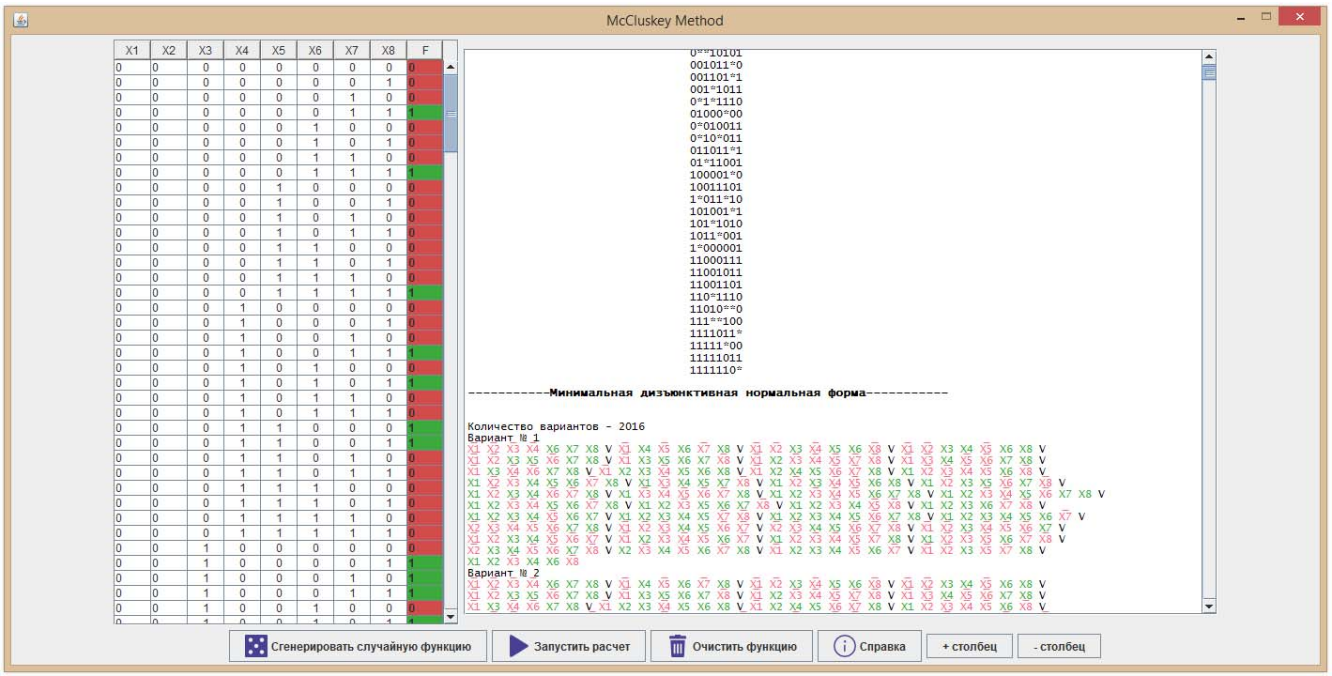


Рис. 3. Пример работы программы

Реализация

Программа состоит из двух классов:

• Value – основной класс, в котором происходят все вычисления, обработка ввода и вывода на экран;

• Cub – это специальный класс, в который входит динамический массив, реализующий импликанты и переменные типа Boolean, используемый для понимания, использована (вычеркнута) ли данная импликанта в конкретной ситуации.

Разработаны следующие методы:

• метод GetStringSDNF() – считывание значений функции из таблицы;

• метод GetFirstCube() – сортировка по количеству единиц;

• метод GetCube() – сравнение соседних строк таблицы. Производится сравнение каждой импликанты первого списка с каждой из второго и т. д. Если они отличаются на одно значение, стоящее в той же позиции, то это значение заменяется символом *;

• метод RDuplicate() – удаление повторяющихся элементов от нового получившегося списка;

• метод TableImplicant() – расчет таблицы импликант;

• метод MatchOneCube() – проверяет, покрывают ли простые импликанты члены СДНФ, если да – ставится метка;

• метод SokrTable() – выполняет расчет покрытиятаблицы импликант без учета ядра. Сначала считается, сколько простых импликант и конституент единицы осталось «невычеркнуыми», т. е. не вошедшими в ядро. Затем из них создается еще одна таблица.

Заключение

Рассмотрены распространенные в изучении методы минимизации логических функций: эквивалентных преобразований, карт Карно, Куайна и Куайна–Маккласки. Произведен их анализ и сравнение, выявлены достоинства и недостатки каждого из них.

Анализ показал, что первый метод – метод эквивалентных преобразований – в связи с неалгоритмичностью сложно запрограммировать. Метод карт Карно основан на зрительном анализе, поэтому данный метод с трудом может быть применен для реализации вычислительной техникой. В методе Куайна при конструировании множества всех простых импликант тратится много времени на поиск соседних элементарных конъюнкций в преобразуемой ДНФ, что занимает значительное время. Усовершенствование Маккласки – разбиение конституент на группы по количеству в них единиц – систематизировало метод Куайна, значительно сократило число сравниваемых двоичных номеров при склеивании, что позволило уменьшить время на выполнение минимизации.

На основе этих выводов был выбран метод Куайна–Маккласки для разработки Java-приложения, реализующего минимизацию логических функций в случае произвольного числа переменных.

Литература

1. Лихтарников Л.М. Математическая логика / Л.М. Лихтарников, Т.Г. Сукачева. – СПб. : Лань, 2008. – 288 с.

2. Глухов М.М. Математическая логика / М.М. Глухов, А.Б. Шишков. – СПб. : Лань, 2012. – 405 с.

3. Закревский А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Л.Д. Поттосин. – М. : Физматлит, 2007. – 584 с.

4. Набебин А.А. Дискретная математика / А.А. Набебин. – М. : Научный мир, 2010. – 509 с.

5. Ершов Ю.Л. Математическая логика / Ю.Л. Ершов, Е.А. Палютин. – СПб. : Лань, 2005.

6. Курош А.Г. Лекции по общей алгебре. – СПб. : Лань, 2002. – 396 с.

7. Карри Х. Основания математической логики. – М. : Мир, 1966.

8. Victor P. Nelson, H. Troy Nagle, Bill D. Carroll, David Irwin, Digital Logic Circuit Analysis and Design, 2 Ed Prentice Hall. 1995. – ISBN 0134638948. – 842 pages.

9. Bradford Henry Arnold. Logic and Boolean Algebra. – Dover Publications Reprint, 2011. – 158 pages.

10. Goodstein R.L. Boolean Algebra // The Commonwealth and International Library of Science, Technology, Engineering and Liberal Studies, Vol. 6, Pergamon Press, Oxford, 1963. – 140 pages.

11. H. Graham Flegg. Boolean Algebra. – John Wiley & Sons Canada, Limited, 1964. – 261 pages.

Choosing an Optimal Method of Minimization while Developing a Program for Searching Minimal Disjunctive Normal Forms

V.A. Tabanina
Emperor Alexander I St. Petersburg State Transport University
St. Petersburg, Russia
alekseevnav@gmail.com

Abstract. **The methods of minimizing logical functions, equivalent transformations, Carnot, Quine and Quine-McCluskey maps, are widely used in the study. They are analyzed and compared, the main mathematical relationships underlying the methods, advantages and disadvantages are indicated. An optimal method is chosen with the goal of developing a Java application that minimizes the logical functions in the case of an arbitrary number of variables.**

Keywords: **minimization, boolean algebra, disjunctive normal form, canonical normal form, minimization of the logic functions, equivalent transformations, Carnot's cards, Quine's method, Quine–McCluskey method.**

REFERENCES

1. Likhtarnikov L.M. Logic Theory [Matematicheskaya Logika], SPb. : Fallow deer, 2008. – 288 pages. (In Russ.)
2. Glukhov M.M. Logic theory [Matematicheskaya Logika], SPb. : Fallow deer, 2012. – 405 pages. (In Russ.)
3. Zakrevsky A.D. Logical bases of design of the discrete devices [Logicheskie osnovy proektirovaniya diskretnykh ustroystv], D. Zakrevsky, L.D. Pottosin. – Moscow : Fizmatlit, 2007. – 584 pages. (In Russ.)
4. Nabebin A.A. The discrete mathematics [Diskretnaya matematika], Moscow: Scientific world, 2010. – 509 pages. (In Russ.)
5. Yershov Yu. L. Logic theory [Matematicheskaya logika], Yu.L. Yershov, E.A. Palyutin. – SPb. : Fallow deer, 2005. (In Russ.)
6. Kurosh A. G. Lectures on the general algebra [Lektsii po obshchey algebre.]. – SPb. : Fallow deer, 2002. (In Russ.)
7. Carry X. Bases of a logic theory [Osnovaniya matematicheskoy logiki]. – M. : Patterns, 1966. (In Russ.)
8. Victor P. Nelson, H. Troy Nagle, Bill D. Carroll, David Irwin, Digital Logic Circuit Analysis and Design, 2 Ed Prentice Hall. – 1995. – ISBN: 0134638948. 842 pages.
9. Bradford Henry Arnold. Logic and Boolean Algebra. – Dover Publications Reprint, 2011. – 158 pages.
10. Goodstein, R. L., Boolean Algebra // The Commonwealth and International Library of Science, Technology, Engineering and Liberal Studies, Vol. 6, Pergamon Press, Oxford, 1963, 140 pages.
11. H. Graham Flegg. Boolean Algebra. – John Wiley & Sons Canada, Limited, 1964. – 261 pages.