

$$\bar{\mu} = \frac{1}{n} \sum_{i=1}^n \mu_i; \quad (10)$$

$$\mu_i^{(0)} = \int_0^{L_i} y_i I'_i(y_i) dy_i; \quad (11)$$

$$\mu_i^{(L_i)} = \int_{L_i}^{+\infty} y_i I'_i(y_i) dy_i; \quad (12)$$

Итоговое значение ранга  $K_i$  узла  $i$  вычисляется по формуле:

$$K_i = \frac{1}{R_i C_{r_{abs}}}. \quad (13)$$

Множество рассчитанных рангов узлов используется для оценки топологии в целом. Определенные в п. 2 условия отнесения топологии к разряду неудовлетворительных могут быть оценены с помощью  $K_i$ .

Наличие в сети «бутылочных горлышек» приводит к прохождению через один ключевой узел большого потока данных – через элемент, являющийся связующим звеном между сегментами сети идет трафик обоих сегментов, что приводит к возрастанию коэффициента нагрузки  $R_i$ . В такой ситуации  $K_i$  значение которого, согласно (13), обратно пропорционально  $R_i$ , будет уменьшаться.

Высокая нагрузка на критичный элемент ведет к возрастанию  $R_i$  и  $C_{r_{abs}}$ , что также приводит к уменьшению  $K_i$ .

Вследствие объединения нескольких критичных элементов в один сегмент сети большая часть критичного трафика проходит через один узел, увеличивая тем самым его собственную критичность.

На выходе модели формируется распределение  $K_i$  по элементам сети, на основе среднего значения по сети может определяться порог отказоустойчивости  $K_{min}$ , значения ниже которого говорят о низком

качестве топологии за счет неоптимального распределения нагрузки и расположения высококритичных узлов.

Программная реализация оценки топологии сети

Была осуществлена программная реализация модели. На вход данной программы подается топология сети в виде матрицы смежности, статистика, использования элементов, характерная для конкретного состава семьи (на данном этапе бралась статистика использования приборов для одного жильца), значения критериев оценки критичности для каждого элемента. На выходе программа выдает значение ранга  $K_i$  для каждого узла, на основании которых можно судить о степени отказоустойчивости рассматриваемой топологии.

#### Заключение

В результате работы была разработана математическая модель оценки отказоустойчивости сети, основанная на определении степеней критичности ее элементов и формировании оптимальной топологии при заданном распределении нагрузки. Данная модель позволяет автоматизировать анализ топологии сети системы «Умный дом», сформировать оптимальное распределение элементов в сети, позволяющее предотвратить нарушения безопасности данного типа систем.

Дальнейшим этапом данного исследования будет сбор и анализ статистики использования элементов сети для различного состава семей, а также интеграция ее с данной моделью. Основываясь на реальных данных, можно будет более точно сформировать прогноз нагрузки на каждый элемент.

#### Список литературы

1. «Умный дом» – маркетинговое исследование российского рынка: текущее состояние и прогноз развития. [Электронный ресурс] – URL: [http://www.directinfo.net/index.php?option=com\\_content&view=article&id=139%3A2010-07-06-13-57-09](http://www.directinfo.net/index.php?option=com_content&view=article&id=139%3A2010-07-06-13-57-09) (дата обращения 01.03.2016).
2. Аналитический отчет о ключевых тенденциях в сфере информационной безопасности [Электронный ресурс] – URL: <https://www.esetnod32.ru/company/press/center/eset-2014-god-prineset-bum-tekhnologiy-anonimnosti/> (дата обращения 01.03.2016).
3. Обзор систем и технологий «Умный дом» [Электронный ресурс] – URL: <http://www.a3d.ru/design/tehnolog/25> режим доступа: свободный (дата обращения 01.03.2016).

### Секция «Информационные технологии в IT-индустрии, образовании, управлении и науке», научный руководитель – Сенкевич Л.Б., канд. пед. наук

#### ЧИСЛОВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ В ОДНОМЕРНОМ МАССИВЕ

Гагарина А.А.

ТюмГНГУ, Тюмень, e-mail: [nastyga.gagarina.2016@mail.ru](mailto:nastyga.gagarina.2016@mail.ru)

Массив – это числовая последовательность однотипных данных которые хранятся в памяти компьютера, каждый элемент классифицируется порядковым номером (индексом) элемента. Порядковый номер элемента массива носит название индекса этого элемента. В числовых массивах как значения применяются – числа. Массивы с числовым массивом дают возможность применить более простой способ наборов данных, так как для того чтобы перейти к следующему значению необходимо лишь увеличить на единицу индекс предыдущего значения данного массива.

Массив – это совокупность данных, которые выполняют подобные функции, и классифицируются одним именованием. В случае если за каждым таким элементом массива будет только один его порядковый номер, то подобный массив будет называться линейным, или же одномерным.

Пример: числовая последовательность четных натуральных чисел 2, 4, 6, ..., N представляет собой линейный массив, составляющие которого можно

назвать  $A[1]=2, A[2]=4, A[3]=6, \dots, A[K]=2*(K+1)$ , где K – номер элемента, а 2, 4, 6, ..., N – значения. Индекс (порядковый номер элемента) записывается в квадратных скобках после имени массива. К примеру,  $A[7]$  – 7-й элемент массива A;  $D[6]$  – 6-й элемент массива D.

Создание массива. Для создания массива есть три основных способа: присваивание значения одному из составляющих элементов будущего массива, использование аргумента `array()`, главной чертой, которой является то, что она возвращает массив в качестве своего значения.

#### Непосредственное присваивание

Простой способ создания массива заключается в выполнении с некоторой переменной таких действий, якобы эта переменная уже представляет собой массив, и присваивании ей значения. Для создания данного массива необходимо определить значения его элементов и индексов. В качестве таких элементов могут применяться любые значения, к примеру строки, числа и другие значения.

#### Система `array()`

Данный метод массива заключается в употреблении такой конструкции, как `array()`, данная система создает массив на основании перечислении элементов.

Выводом становится, что такие элементы в массиве данной системы сохраняются в указанном порядке.

#### Функции, возвращающие массивы

Данный способ происходит в вызове функции, которая возвращает массив. К примеру, в виде динамически создаваемых массивов возвращают собственные результаты почти все функции, обеспечивающие взаимодействие с базой данных.

#### Удаление элементов из массивов

Операция удаление элемента из массива реализуется очень просто, подобно операции удаления значения, которое присваивается переменной. Достаточно просто вызвать функцию unset().

#### Методы доступа к элементам

Доступ к элементу одномерного массива осуществляется при помощи конструкции имя\_массива [индекс], причем эту систему можно использовать в выражениях (тогда берется значение указанного элемента массива) или же в левой части операции присваивания (тогда обозначенному составляющей массива присваивается значение выражения, который стоит в правой части этой операции присваивания).

Также необходимо помнить следующие правила для доступа к элементам массива в C++:

- 1) в C++ индексы всех массивов всегда начинаются с 0, и должны быть целыми числами;
- 2) C++ не проверяет корректность индекса, т. е. тот факт, что индекс располагается в разрешенных границах (от 0 до количества составляющих массива-1). Потому что образ char считается целочисленным, допустимо брать переменные и константы такого типа в качестве индексов, однако нужно помнить, что на некоторых компиляторах для символов, не входящих в набор ASCII, эти значения могут быть отрицательными.

Ввод (вывод) составляющих массива случается при применении алгоритмов повторяющейся структуры с указанным количеством повторений, в которых численность вложенных друг в друга циклов ориентируется размерностью массива, а численность изменений переменных каждого из циклов – очень максимально вероятным численностью составляющих массива в предоставленном измерении.

Ввод-вывод составляющих одномерного массива случается при использовании алгоритмов с обозначенным числом повторений, которые станут равны численности членов массива, а вслед за этим в цикле случается поэлементный ввод (вывод) значений всякого из членов массива.

Таким образом, можно сказать что числовая последовательность одномерного массива в паскале – это база данных, каждая из которых содержит определенный порядковый номер для обращения к нему. Применяется массив для хранения числовых значений в определенном количестве. Одномерным массивом очень удобно обрабатывать огромное количество однотипных данных.

#### Список литературы

1. [http://fullref.ru/job\_32fb0120b89472c0eb93377dafc49fce.html].
2. [http://addphp.ru/materials/base/1\_11.php].

#### НЕЙРОСЕТЕВЫЕ МЕТОДЫ РЕАЛИЗАЦИИ АДАПТИВНОГО ТЕСТИРОВАНИЯ

Макарова К.И., Кудрина Е.В.

*Саратовский национальный исследовательский государственный университет, Саратов, e-mail: kmakarova1994@gmail.com*

Развитие тестирования как метода психолого-педагогической диагностики началось в XIX веке. В настоящее время тестирование интенсивно применяется в педагогических целях, в том числе и для диагностики результатов учебной деятельности студентов вузов [1].

Одной из проблем в развитии тестирования является недостаточная надёжность оценок, получаемых с помощью распространённых моделей тестирования знаний. Решить эту проблему позволяет использование адаптивного тестирования

Адаптивное тестирование (АТ) – разновидность тестирования, при котором порядок предъявления заданий (или трудность заданий) зависит от ответов испытуемого на предыдущие задания. Адаптивное тестирование позволяет повысить эффективность контрольно-оценочных процедур за счет индивидуализации процедуры тестирования, что, в свою очередь, приведет к точности измерения, минимизации числа заданий и времени на контроль.

Для организации адаптивного тестирования необходимо разработать следующие компоненты: цели тестирования, способы построения набора заданий тестирования, методы проведения тестирования, методы проверки результатов тестирования, методы оценивания результатов тестирования, правила окончания тестирования. Особый интерес представляют методы проведения тестирования, классификация которых приведена в работе [2].

На данном этапе проведен анализ исследований в области теории и практики АТ [3-5], который позволяет сделать вывод о том, что способы построения траектории АТ при помощи задания переходов между состояниями (используя Байесовские сети, цепи Маркова, сети Петри, конечные автоматы) достаточно хорошо изучены. В связи с этим была определена цель работы – рассмотреть возможность использования нейросетевых методов для построения траектории АТ.

Процесс тестирования с использованием нейронной сети можно описать следующим образом [6]: На вход подаются 3 параметра:  $X_1$  – номер этапа тестирования,  $X_2$  – уровень сложности вопроса в тесте,  $X_3$  – количество правильных ответов, набранных испытуемым после одного этапа тестирования. Выход нейронной сети –  $Y_i$  – данные о повышении или понижении уровня сложности вопроса на следующем этапе тестирования испытуемого, где  $i = 1, N$ . Где  $N$  – количество этапов тестирования.

На начальной фазе тестирования испытуемому предлагается пройти первый этап, вопросы которого отражают фундаментальные знания по дисциплине. Целью данного этапа является выявление уровня подготовленности испытуемого. В процессе тестирования система переводит испытуемого с одного этапа тестирования на другой с учетом его подготовленности, повышая или понижая уровень сложности вопросов в тесте. Процесс завершается по прохождению испытуемым всех запланированных в тестировании этапов, подведением итоговой оценки.

В дальнейшем планируется изучить программное обеспечение, применяемое для моделирования нейронных сетей, что позволит разработать тестирующую систему, использующую нейросетевые методы для построения траектории АТ.

#### Список литературы

1. Балакирева Е.И. Использование Moodle для дистанционного тестирования учебных достижений студентов вузов/ Е.И. Балакирева, Е.В. Кудрина // Материалы Международной научной конференции, посвященной 100-летию СГУ «Компьютерные науки и информационные технологии». – Саратов: Изд-во Саратов. ун-та, 2009. – С. 19-22.
2. Метод разработки алгоритмов адаптивного тестирования [Электронный ресурс]. – URL: http://cyberleninka.ru/article/n/metod-razrabotki-algoritmov-adaptivnogo-testirovaniya (дата обращения: 28.11.2015).
3. Юрьев Г.А. Математическая модель интерпретации результатов компьютерного тестирования с использованием Марковских сетей: Автореф. дис. ... канд. физ.-мат. наук. – М., 2013.
4. Ульянов Д.А. Марковская модель адаптивного тестирования и ее программная реализация в условиях дистанционного обучения: Автореферат дис. ... канд. техн. наук. – Иркутск, 2005.