Университет „Проф. д-р Асен Златаров“ – гр. Бургас

СОФТУЕР ЗА ПРИЛАГАНЕ НА ИНТЕРКРИТЕРИАЛЕН АНАЛИЗ

Потребителска и техническа документация

Деян Г. Мавров
Ноември 2017 г.

Работата по приложението беше извършена в рамките на проект ДФНИ-И02/5 – "Интеркритериален анализ – нов подход за вземане на решения".

За контакти:

Университет „Проф. д-р Асен Златаров"
Катедра „Компютърни и информационни технологии"
Гр. Бургас, бул. „Проф. Яким Якимов" № 1
e-mail: deyan_mavrov@btu.bg

Съдържание

Указания за употреба

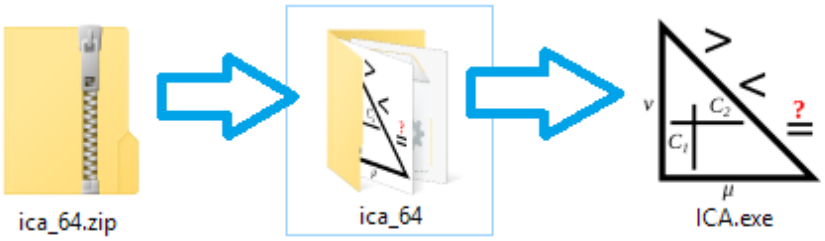
1. Инсталация

За да получите програмата за Интеркритериален анализ (наричана от тук нататък за краткост в тези инструкции просто „програмата“) посетете сайта на проекта “Интеркритериален анализ – нов подход за вземане на решения“ на адрес: www.intercriteria.net. Там можете да намерите и други програми за прилагане на Интеркритериалния анализ, както и допълнителна информация за работата по проекта.

Ако искате да прегледате изходния код на програмата, може да посетите следния адрес: http://bitbucket.org/intercriteria/icdm/.

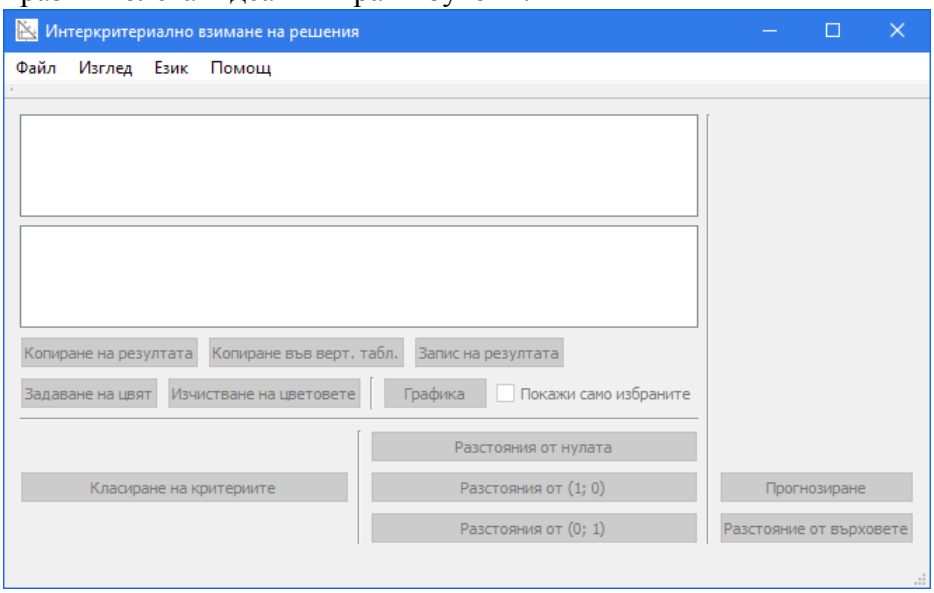
Програмата се разпространява под формата на ZIP архив, в който са събрани изпълнимия файл и всички необходими библиотеки. Програмата има две версии за Windows – 32-битова и 64-битова. Името на архива с 32-битовата версия завършва на „_32.zip“, а на този със 64-битовата версия – на „_64.zip“. Трябва да използвате тази версия, която съответства на архитектурата на вашата операционна система.

За да използвате програмата просто разархивирайте цялото съдържание на архива в нова папка. След това влезте в нея и потърсете файла „ICA.exe“ и щракнете двукратно върху него.



2. *Прочитане на данните*

При стартиране приложението показва на екрана прозорец с празни полета и деактивирани бутони.



За да се започне работа първото, което трябва да се избере, е името на файла с входните данни, от който се взима информация за оценките на даден брой обекти по даден брой критерии.

За да успее програмата да прочете файла, той трябва да бъде един от следните видове:

- работна книга, поддържана от Microsoft Excel
- чист текстов файл с елементи, разделени от табулации

За да може програмата да чете работни книги, поддържани от Excel, е необходимо на компютъра да бъде инсталиран Microsoft Office. Системата е изпробвана с Office 2007 и следващи версии, но би трябвало да работи и с Office 2003.

За отваряне на текстови файлове, разделени с табулации, не е необходимо да се инсталира допълнителен софтуер. Важно е да се спомене, че програмата очаква дробните числа в текстовия файл да са с десетичния разделител, посочен в системните настройки (за България обикновено запетая (**,**), а за Северна Америка – точка (**.**)).

Ако за прочитането използваме Excel, данните трябва да са поставени в един от работните листи на книгата, като започват от първи ред, първа колона. По подразбиране се чете от първия работен лист.

При прочитане от текстов файл, разделен с табулации, данните трябва да започват от първия ред на файла.

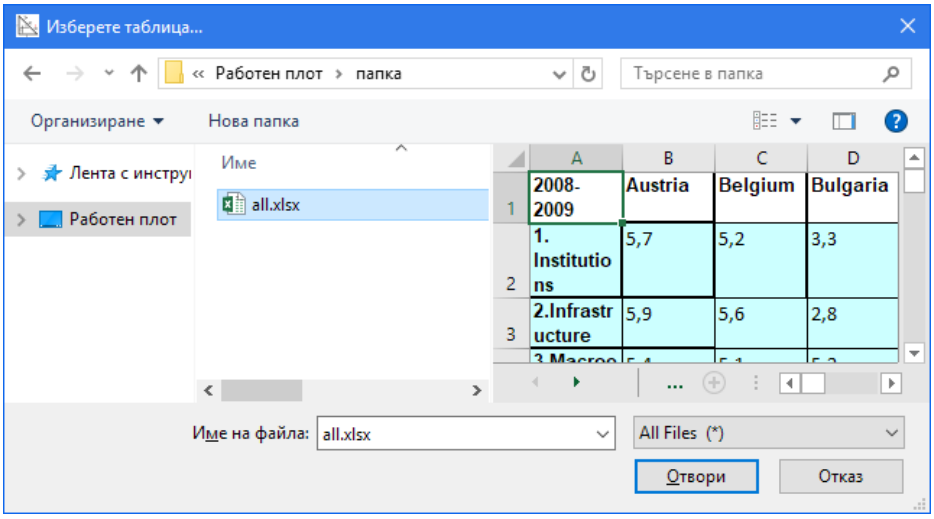
И при двата варианта данните трябва да са подредени по следния начин:

	A	B	C	…	*z-1*	***z***
1	*Не се използва*	Име на обект 1	Име на обект 2	…	Име на обект [z-2]	
2	Име на критерий 1.1	Оценка К1.1-О1	Оценка К1.1-О2	…	Оценка К1.1-О[z-2]	
3	Име на критерий 1.2	Оценка К1.2-О1	Оценка К1.2-О2	…	Оценка К1.2-О[z-2]	
…	…	…	…	…	…	…
a-1	Име на критерий 1.[a-2]	Оценка К1.[a-2]-О1	Оценка К1.[a-2]-О1	…	Оценка К1.[a-2]-О[z-2]	
a				…		
a+1	Име на критерий 2.1	Оценка К2.1-О1	Оценка К2.1-О2	…	Оценка К2.1-О[z-2]	
a+2	Име на критерий 2.2	Оценка К2.2-О1	Оценка К2.2-О2	…	Оценка К2.2-О[z-2]	
…	…	…	…	…	…	…
x-1	Име на критерий 2.[x-1-a]	Оценка К2.[x-1-a]-О1	Оценка К2.[x-1-a]-О2	…	Оценка К2.[x-1-a]-О[z-2]	
x				…		

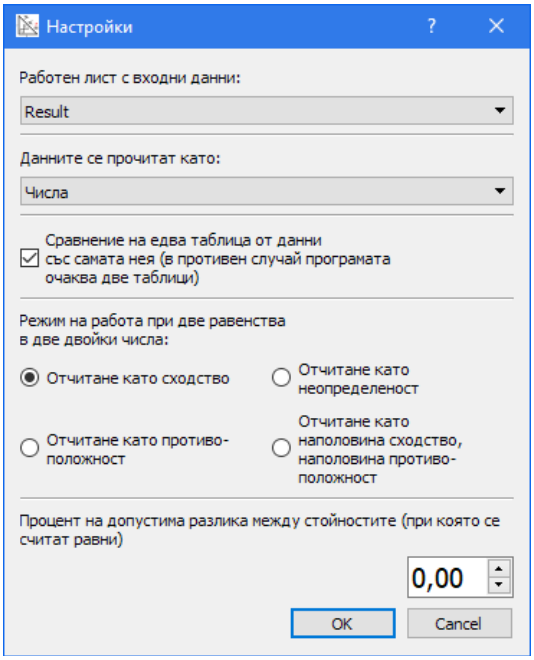
Това са две таблици, разделени от празен ред. Имената на критериите са в първата колона, а имената на обектите – в първия ред, като последните не се повтарят за втората таблица.

Ако във входния файл са налични както в схемата две таблици, в които критериите са различни, но оценяваните обекти са същите, то може да се сравнят критериите от първата таблица с тези от втората. По избор потребителят може да реши да използва само една таблица (т.е. ако във файла има две – първата от тях), при което критериите от нея се съпоставят един с друг (вж. по-долу).

Обикновено избор на файл се прави като се избере менюто Файл→Отваряне. В по-новите издания на Windows е възможно в този прозорец да се направи и предварителен преглед на избрания файл, преди да се отвори.



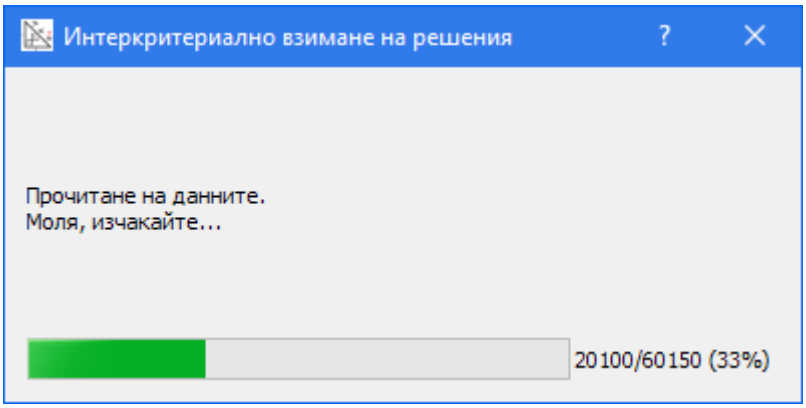
След избирането на файл се отваря прозорец за конфигурация.



В него могат да бъдат настроени следните възможности:

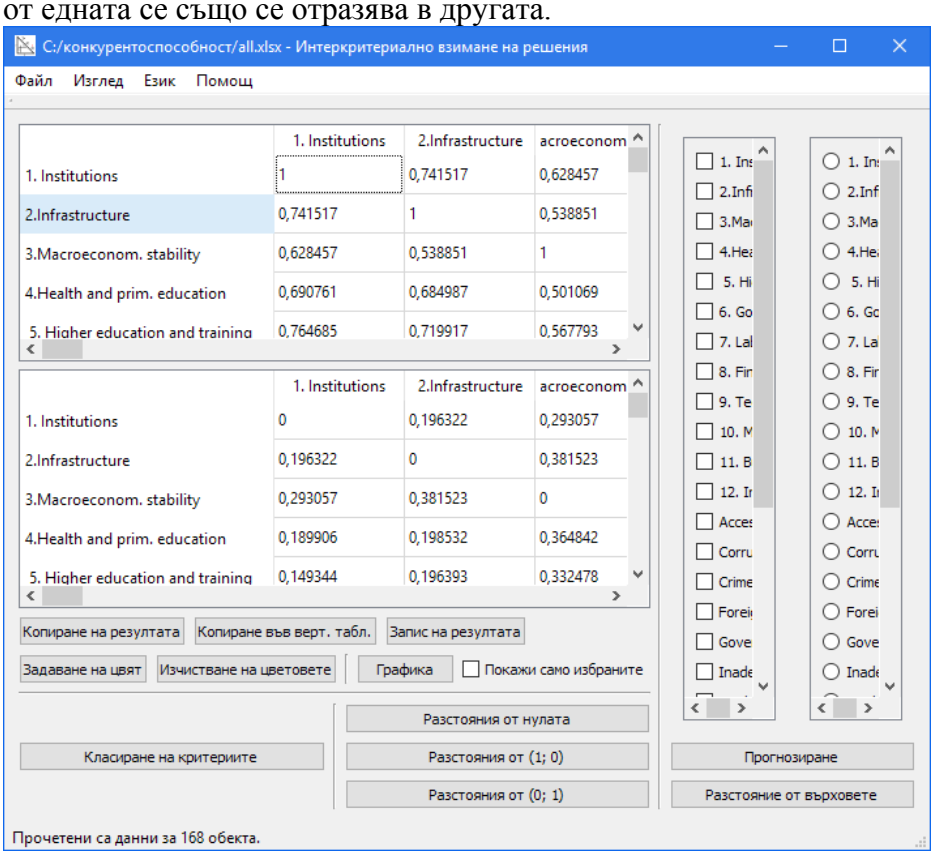
- Избор на работен лист, от който да се прочетат входните данни (по подразбиране – първият лист). Ако входният файл е текстов, тази опция е неактивна.
- Дали стойностите от входната матрица да се четат като числа или като стойности +/- (при изследване на системи от логически аксиоми и др.). Това определя дали да се използва стандартният вариант на Интеркритериалния анализ или този с пряко сравнение на стойности.
- Дали да се очаква наличието на две входни таблици или да се сравнява една входна таблица със себе си.
- Как да се третира наличието на две равенства при сравняването на релациите между две двойки стойности.
- Задаване на процент на толеранс, в чиито рамки ще се приема, че две стойности са равни.

След натискане на бутона „OK“ започва изчислението на резултата. Програмата изписва някои съобщения за статуса на изчислението.



След завършване на изчислението, резултатите от Интеркритериалния анализ се изобразяват в двете таблични полета на главния прозорец на програмата. Горната таблица изрежда стойностите на принадлежността (μ), а долната на непринадлежността (ν) от интуиционистки размитите двойки, които се получават като резултат от анализа за всяка двойка критерии. Двете таблици са свързани – плъзгането в едната се

отразява веднага в другата и изборът на двойка критерии (клетка) от едната се също се отразява в другата.



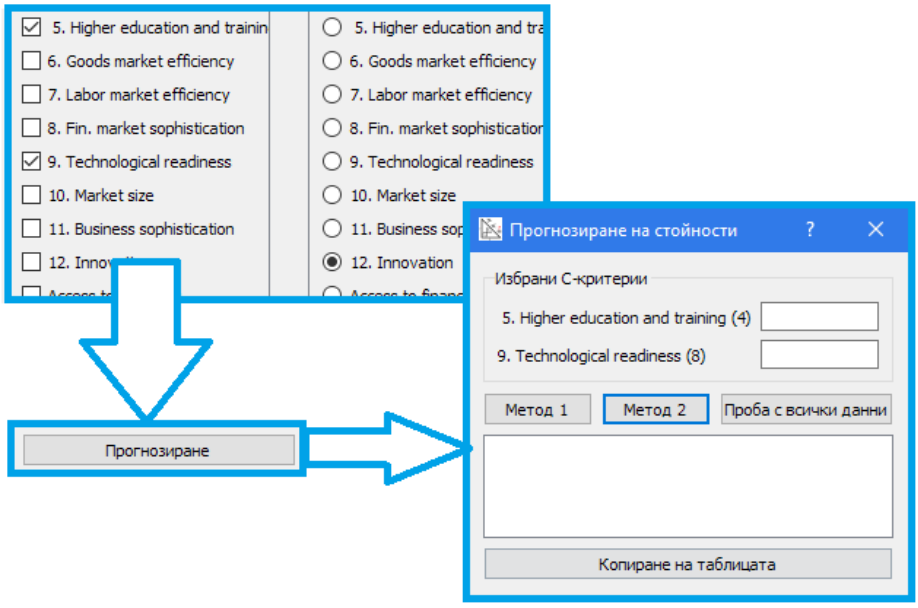
След като вече резултатът е получен, програмата предоставя някои допълнителни възможности за допълнителния му анализ.

3. *Използване на резултатите за прогнозиране на стойности*

Вдясно от табличните полета се намират две списъчни вертикални менюта. Лявото от тях дава възможност да се изберат един или няколко критерия (C-критерии), оценките спрямо които ще използваме при предсказване на стойности. От менюто вдясно може да се избере само един критерий (D-критерий), за който ще се опитваме да прогнозираме оценката за обект/обекти, въз основа на оценките, налични за избраните вляво отправни критерии.

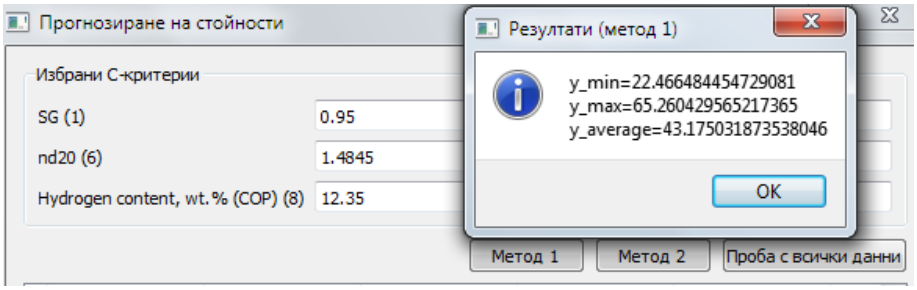
Тук трябва да изберем тези C-критерии, които показват най-силна корелация с D-критерия.

При натискане на бутон „Изпълнение“ се отваря прозорецът за прогнозиране на стойности. Тук може да се работи с двата метода за прогнозиране, описани в първата публикация за Интеркритериалния анализ [1].

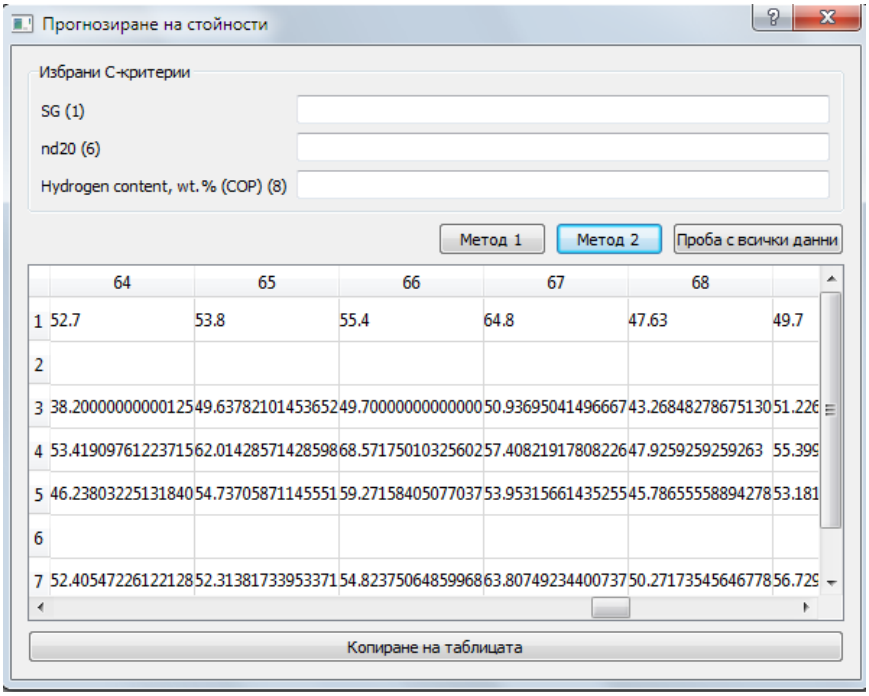


Дадени са две възможности за работа:

- Прогнозиране на оценката по D-критерия за един нов обект. За целта се въвеждат числа за оценките по C-критериите в съответните текстови полета, избира се един от двата метода и се натиска съответно бутон „Метод 1“ или „Метод 2“, след което при успех се отваря диалогов прозорец с резултатите от съответния метод.

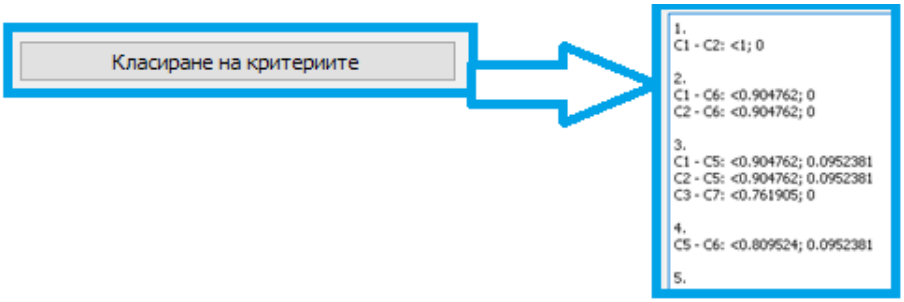


- Пробно прогнозиране по всички налични обекти и по двата метода – бутон „Проба с всички данни“. За целта се взимат един по един наличните обекти и за всеки от тях се зачитат само оценките по С-критериите, а оценката по D-критерия се пренебрегва. В табличния обект в долния край на прозореца за прогнозиране на стойности се формира таблица, която съдържа реалните оценки за всеки обект по D-критерия, а под тях – прогнозираните стойности по всеки от двата метода. Целта е да се провери доколко успешно методите прогнозират стойности по избраните С-критерии за избрания D-критерий.



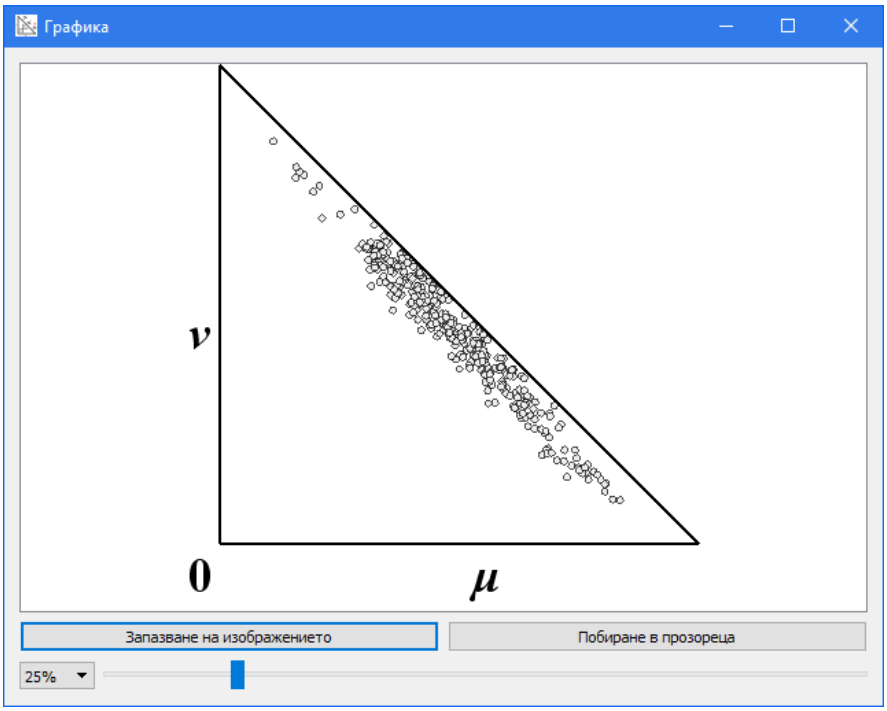
4. Класиране на двойките критерии

Един от методите за класиране на двойките критерии по получените резултати от анализа е този, използващ положението на съответстващите им интуиционистки размити (ИР) точки в ИР триъгълник спрямо точката (1; 0) [2]. Бутонът „Класиране на критериите“ изпълнява класирането, след което резултатът се извежда в текстово поле в нов прозорец.



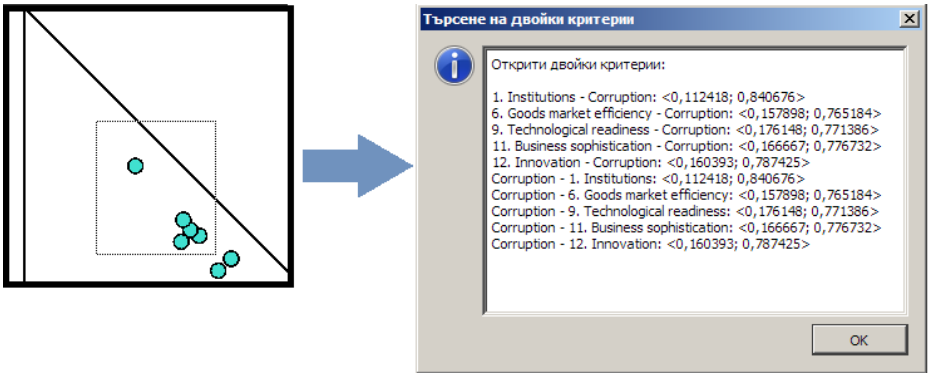
5. *Графично изобразяване на резултатите*

Програмата може да генерира графика на получените ИР двойки в интуиционистки размития триъгълник (с върхове (0; 0), (1; 0) и (0; 1)). При натискане на бутона „Графика“ се отваря нов прозорец, където всяка ИР двойка е поставена като точка в триъгълника.

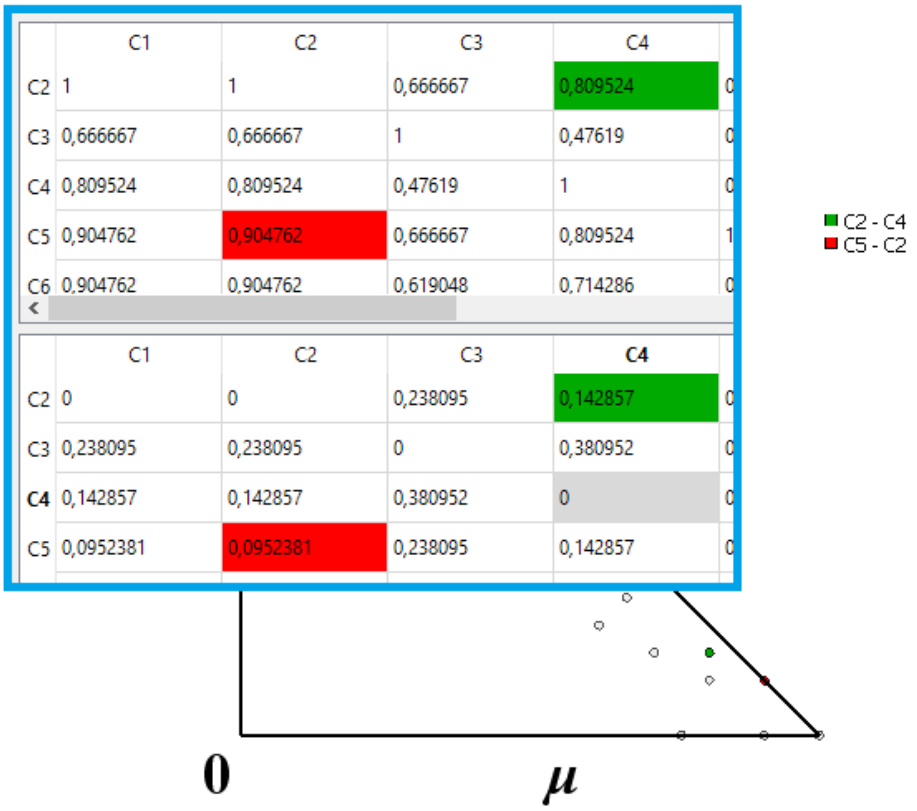


Прозорецът позволява мащабиране и запис на графиката като файл с изображение.

Ако искаме да проверим кои двойки критерии формират дадена точка или област от точки, е достатъчно да изрисуваме правоъгълник около набелязаните точки, след което да отпуснем мишката. След това се показва прозорец с имената на откритите двойки критерии и стойностите на техните показатели.



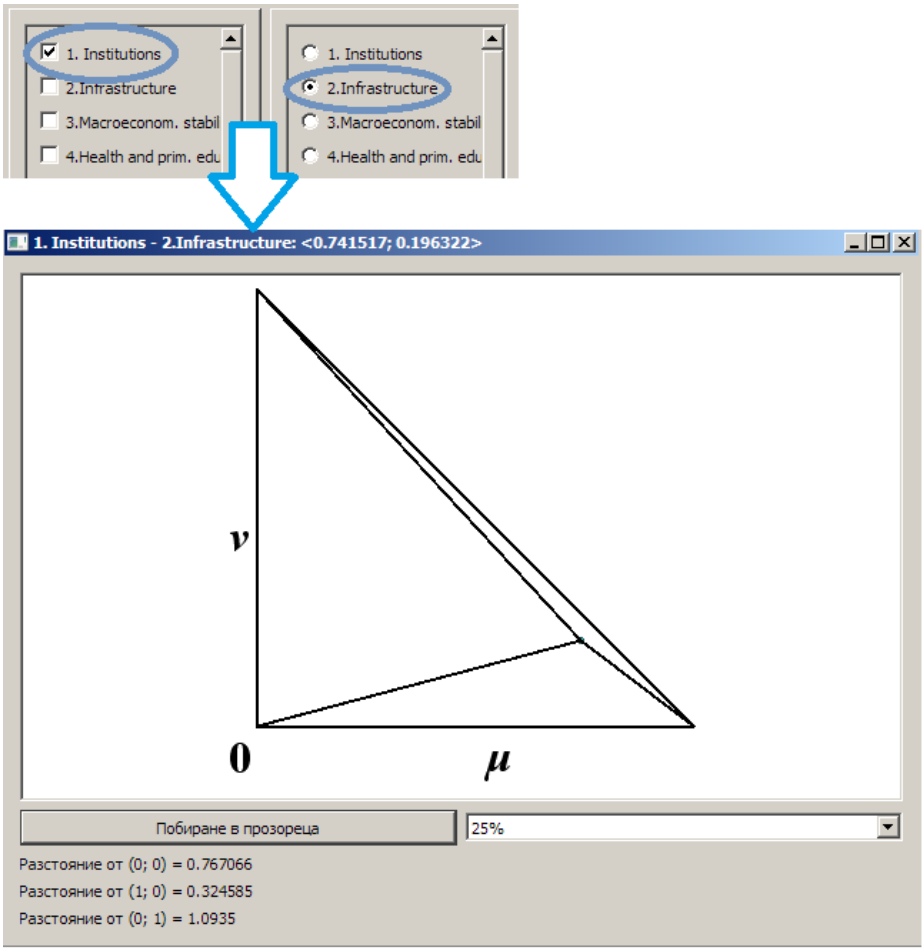
Програмата предоставя и начин да се открояят избрани точки в триъгълника. За да се използва тази функция, потребителят трябва да щракне двукратно върху една от клетките в коя да е от двете таблици с резултата, след което да избере цвят в появилия се прозорец. Тъй като и принадлежността (горе) и непринадлежността (долу) се отнасят до една и съща двойка критерии, съответстващата клетка в срещуположната таблица също получава избрания цвят.



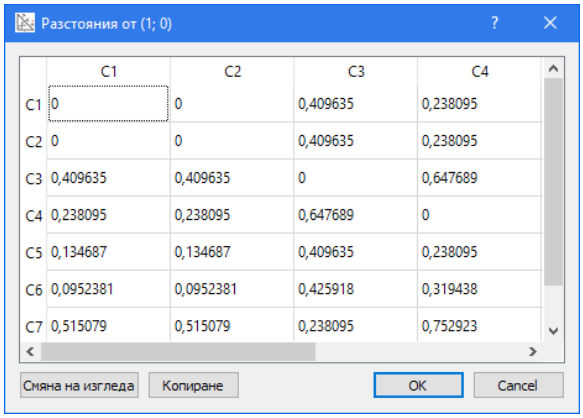
При маркиране на полето „Покажи само избраните“ в графиката се изобразяват единствено отбелязаните с цвят двойки критерии. Цветът на дадена клетка може да бъде променен с двойно щракване върху нея. Ако искаме няколко клетки да имат един и същ цвят е необходимо само да ги маркираме с показалеца и да натиснем бутона „Задаване на цвят“. С бутона „Изчистване на цветовете“ всички цветове се заменят с неутралния (бял), който кара съответната точка да се изобрази като обикновена в графиката.

В същия изглед можем да изобразим стойността за конкретна двойка критерии с отстоянията ѝ от върховете на триъгълника. За целта трябва да се маркира един критерий от левия списък (вж. точка 2) и един от десния (при повече маркирани

критерии в левия списък се зачита първия). Натиска се бутонът „Разстояние от върховете“. Под графиката на избраната двойка се изписват числовите стойности на отстоянията.



Ако пък има необходимост да се видят отстоянията на всички точки от някой от трите върха, могат да се използват бутоните „Разстояние от нулата“, „Разстояния от (1; 0)“ и „Разстояния от (0; 1)“. При натискане на един от тях се отваря прозорец с таблица, в която са изписани отстоянията. Таблицата може да бъде копирана.



6. Запис и прехвърляне на резултатите

Програмата позволява резултатът да бъде записан (бутон „Запис на резултата“). Мястото на запис зависи от видът на отворения файл.

Ако входният файл е отворен с помощта на Excel, то резултатът се записва в нов лист, който се вмъква в работната книга веднага след листа, в който са входните данни.

Ако файлът е текстов, разделен с табулации, то резултатът се записва в нов файл. Ако входният файл има име „input.txt“, то изходният ще има име „input.txt.out.txt“. Ако файлът съществува, съдържанието му се изтрива и се заменя с новото.

При всеки опит за запис на резултата, програмата пита потребителя дали наистина иска да извърши записа, за да не се получат нежелани листове или да се презапише по грешка стар текстов файл.

Бутонът „Копиране на резултата“ позволява двете таблици с числа да се прехвърлят директно в други програми. Данните в тях се прехвърлят по стандартен начин в текстов вид към системния клипборд и от там могат да бъдат поставени в повечето системи за електронни таблици и други приложения.

Технически подробности

1. Използвани технологии

Програмата е реализирана на езика C++ с помощта на софтуерния пакет (фреймуърк) Qt (www.qt.io). За реализация на графичния интерфейс е използван модулът Qt Widgets, който е класическият начин да се разработват графични приложения с Qt. Qt предоставя много класове, част от които служат за разработка на графични прозорци по начин, независещ от операционната система, докато други създават абстракция над системните възможности за работа с процеси, мрежови комуникации, триизмерна и двуизмерна графика и т.н. В кода на програмата са използвани някои езикови конструкции, добавени в C++11.

Основна среда за разработка на програмата са Windows 7 и Windows 10 с компилатора на Microsoft Visual Studio 2013 (CL.exe), като проектирането и съставянето на кода са извършени с Qt Creator 3 и 4. По време на разработката (от края на 2013 г. до сега) програмата е компилирана с различни версии на Qt 5.x, като изходните файлове са кодирани в UTF-8.

Програмата е изпълнявана пробно на версии на Windows от XP нататък, макар че изпълнението в XP може да изисква някои дребни модификации в изпълнимия файл (смяна на минимално необходимата версия на ОС на 5.1, смяна на функцията „GetTickCount64“ с „GetTickCount“).

Кодът, който се обръща към Microsoft Excel, използва ActiveQt, която установява връзка посредством технологията COM (Component Object Model) на „Майкрософт“. За да може програмата да бъде компилирана по ОС, различни от Windows, частта която зависи от ActiveQt е поставена под препроцесорна директива „#ifdef _WIN32“. Всички останали операции са описани с елементи от стандартната библиотека на C++ и Qt, които не

зависят от ОС. Това позволи програмата да бъде успешно компилирана под Linux (конкретно OpenSUSE) с компилатора G++ от GCC под Qt Creator.

Основния език, на който са написани съобщенията в програмата, е българският. Всички съобщения, които са видими за потребителя, са заобиколени в кода с функцията „tr()“. Това дава възможност те да бъдат преведени на други езици с помощното приложение Qt Linguist. Към момента програмата е преведена на английски език.

2. Инструкции за компилация

За да се компилира обществено достъпния код на програмата е необходимо да се инсталира комплекта за разработка на Qt 5 и съответният компилатор (Visual C++ за Windows и GCC за Linux). За изтегляне на кода е необходима и системата за контрол на версиите Mercurial (hg), макар че BitBucket предлага и ZIP архив на последната редакция. Ако всички необходими програми са на разположение и директориите, в които се намират, са добавени в системната променлива PATH, програмата може да се компилира със следните команди (за прегледност изпълнете в празна директория):

```
hg clone https://bitbucket.org/intercriteria/icdm
mkdir build
cd build
lrelease ../icdm/lang/*.ts
qmake ../icdm
make
```

Друг вариант е кодът на програмата да се постави в нова директория и файлът с разширение „.pro“ да се отвори с Qt Creator. Преди да се компилира, трябва да се генерират езиковите файлове, като в менюто се щракне на:

Tools->External->Linguist->Release Translations

3. Структура на програмата

Програмата се състои от следните класове:

- *MainWindow* – клас на основния прозорец, в който също така се извършват операциите по изчисление на резултата. В него се съхраняват и прочетените и получените стойности, за да бъдат достъпни през цялото време на работа на програмата.
- *Calculations* – клас, който изпълнява изчисленията, необходими за сравнение на две колони обекти. Всяка двойка колони се разпределя в една от заделените работни нишки на процеса. След края на работата си всеки обект на този клас сигнализира *MainWindow*, който изчаква да свърши работата по всички двойки обекти.
- *CalcThread* – клас за създаване на работни нишки. Всяка нишка има отделни броячи, които след приключване на работата се сумират в *MainWindow*.
- *Counter* – клас за брояч. Освен в цели числа, представителите на този клас могат да броят и в части, в зависимост от избрания режим за третиране на двойките равенства в четворките от числа.
- *VectorTableModel* и *DistanceTableModel* – класове-модели, които се използват от табличните полета за преглед от тип *QTableView* за бързо изобразяване на числови данни, съхранени в двумерен вектор. Вторият клас пресмята разстоянията до върховете в движение, за да се спести време.
- *NumberDelegate* – клас за промяна на цвета в клетките от таблицата.
- *AllowedDifferenceDialog* – клас на прозореца за настройки при отваряне на файл.
- *GuessD* – клас на прозореца за прогнози.

- *IFS_Triangle* – клас на прозореца за графика (ИР триъгълник).
- *IFS_GraphicsView* – клас на полето за изчертаване на графиката.
- *ImageSize* – клас на диалога за запис на графиката във файл с изображение.
- *Distances* – клас на прозореца за извеждане разстоянията на точките от върховете не ИР триъгълник.
- *ScrollDialog* – клас на прозореца за изобразяване на резултати в текстово поле.
- *Waiter* – клас за изобразяване на прозорец със съобщения за чакане и лента на прогреса.
- *AxExcel* – клас за обръщане към Microsoft Excel чрез ActiveQt.

4. Общ принцип на работа

4.1.Прочитане на данните от входните таблици

След успешно избиране на файл и настройка програмата започва прочитането на данните. За целта се извиква функцията *OpenWorkbook*. Първото, което тя прави, е да провери типът на отваряния файл.

Ако файлът е с разширение на Excel *(.xls, .xlsx*) се извиква функцията *LoadWorkBookExcel*. В началото тя отваря връзка с Excel като COM сървър и през него отваря файлът и създава обект, сочещ към него. От този обект съответно се извлича обект, сочещ към първия работен лист. Последният дава достъп до функции за прочитане на клетка/и, някои от които се използват нататък в програмата.

Принципът за четене (достъпване) на редовете от данни е да се чете от първата клетка дотогава, докато се достигне празна клетка. Първо отваряме обект от тип „клетка“, като извикваме функцията *Cells(int, int)* за първата клетка от реда, съдържаща число. В тази функция, първият параметър е номерът на реда, а

вторият – номерът на колоната (броенето започва от 1, а не от 0, както е обичайно в C++). От този обект извикваме функцията *End(int)*, за да получим обект за последната непразна клетка от реда. Прочитането на всеки ред става посредством COM функцията на Excel *Range(**address1**:**address2**)*, където ***address1*** и ***address2*** са съответно началната и крайната клетка от реда. Естествено, за тази цел би могла да се приложи и функцията *Cells(int, int)*, но в предходни версии на приложението четенето клетка по клетка се оказа твърде бавно спрямо четенето на област от клетки.

Ако файлът е с разширение *.txt* се прави опит той да бъде прочетен като разделен с табулации документ. За прочитането на файла се създава обект от тип *QFile,* чрез който файлът се чете ред по ред, като всеки ред се разделя спрямо табулациите на списък от стойности. Първият ред съдържа заглавията на колоните, а първият елемент от всеки следващ ред е заглавие на реда. За да се преобразуват както трябва дробите от текст в числов вид се използва функцията *QLocale::decimalPoint()*, която връща настроения в системата знак, използван за десетичен разделител.

Действията и функциите, които използват Excel, са поставени в блокове, оградени с *#ifdef _WIN32 ... #endif*, което гарантира независимостта на програмата от Windows.

Сега ще опишем общия процес на прочитане стъпка по стъпка. Нека отново погледнем диаграмата на формата, който се изисква от входните файлове:

	A	B	C	…	*z-1*	***z***
1	*Не се използва*	Име на обект 1	Име на обект 2	…	Име на обект [z-2]	
2	Име на критерий 1.1	Оценка К1.1-О1	Оценка К1.1-О2	…	Оценка К1.1-О[z-2]	
3	Име на критерий 1.2	Оценка К1.2-О1	Оценка К1.2-О2	…	Оценка К1.2-О[z-2]	
…	…	…	…	…	…	…
a-1	Име на критерий 1.[a-2]	Оценка К1.[a-2]-О1	Оценка К1.[a-2]-О1	…	Оценка К1.[a-2]-О[z-2]	
a				…		
a+1	Име на критерий 2.1	Оценка К2.1-О1	Оценка К2.1-О2	…	Оценка К2.1-О[z-2]	
a+2	Име на критерий 2.2	Оценка К2.2-О1	Оценка К2.2-О2	…	Оценка К2.2-О[z-2]	
…	…	…	…	…	…	…
x-1	Име на критерий 2.[x-1-a]	Оценка К2.[x-1-a]-О1	Оценка К2.[x-1-a]-О2	…	Оценка К2.[x-1-a]-О[z-2]	
x				…		

Първо се прочитат имената на критериите. За всяка от двете таблици се създава масив от тип *vector* (за първата *cTitles*, а за втората – *dTitles*). Спрямо диаграмата, това ще рече следното:

- като се започне от клетка A2 към масива за имена на критерии от първата таблица (*cTitles*) се прибавя съдържанието на всяка клетка надолу по колоната, докато не се стигне до празна клетка (в диаграмата това е клетка A*a*);
- като се започне от първата клетка под достигнатата празна в предната стъпка (в диаграмата – клетка A[*a+1*]) към масива за имена на критерии от втората таблица (*dTitles*) се прибавя съдържанието на всяка клетка надолу по колоната, докато не се стигне до празна клетка (в диаграмата – клетка A*x*).

След това е необходимо да се прочетат всички клетки с числови оценки от двете таблици. За целта в класа на основния програмен прозорец (*MainWindow*) са дефинирани два масива от масиви от дробни числа (*vector<vector<double>>*) с имена *data1* и *data2*. Ако потребителят избере да се търси само една входна таблица, то *data2* става копие на *data1*. Подобно на четенето на имена на критерии, четенето на редове на първата таблица започва от клетка B2 и продължава, докато първата клетка от някой ред не е празна (в диаграмата – клетка B*a*). Отделно за всеки ред четенето продължава надясно, докато не се стигне до празна клетка (в диаграмата – клетка от колона *z*). Броят на пълните клетки на първия прочетен ред служи за норма. Ако бройката на числата в някой от по-долните редове (дори и във втората таблица) е различен, изпълнението се спира. След достигане на края на първата таблица се продължава сходно с втората. Всеки ред се събира в масив от тип *vector<double>*, като след приключване на четенето на реда този масив се прибавя към масивът от редове за съответната таблица (*data1* или *data2*).

При прочитане на стойности от вида „+/-“ всяка клетка се записва в масива като цяло число, като за „+“ стойността е 1, а за „-“ – 2. Всички останали стойности се записват в масивите с 0.

За да се отчетат празните клетки се създава масива *gaps*, където се запомнят позициите на празните клетки за по-нататъшна проверка.

При смяна на един файл с друг всички описани стойности се изчистват и прочитането започва отначало.

4.2.Изчисление на резултата

След като данните от двете таблици са вече налични в съответните масиви може да започне изчислението на образуващите интуиционистки размита двойка стойности на μ и ν за всяка двойка критерии (по алгоритъма, описан в [1]). Те ще се съхраняват в два масива, съответно *Ro* и *Sigma*, които са от тип *vector<vector<double>>* и имат като размерност броя на критериите от първата таблица по броя на критериите от втората. За улеснение при този процес са направени следните приготовления:

- дефинира се класа *Counter*, в който е включен методите *increment* (за увеличаване на брояча), *decrement* (за намаляването му), *reset* (за нулиране) и *value* (за взимане на настоящата му стойност. Броячите от този тип могат да отброяват както в цели числа, така и в половини (0,5).
- декларирана е сравняваща функция *compare*, която приема като параметри две числа. В зависимост от знака, получен при сравнението „<“, „=“ или „>“ връща съответно *1*, *2* или *3*;
- създават се масиви *sum_m_c*, *sum_n_c* и *sum_p_c* със същите размерности както *Ro* и *Sigma*, но от тип *vector<vector<Counter*>>*. В тях се съхраняват броячи на интеркритериалните суми – съответно *sum_m* за броя на съвпаденията на знаците, *sum_n* за несъответствие на знака при сравнение между оценките по всеки два критерия между всеки два обекта и *sum_p* за неопределеност.
- Ако при работа с четворка от стойности една от клетките е празна, това се отчита като неопределеност за тази четворка. За да се постигне това, при наличие на празна клетка функцията *compare* връща 10.
- При работа с входна матрица, където данните са знаци +/-, сравнението не се извършва по четворки от

стойности, както при числовите данни, а по двойки стойности за всяка двойка критерии по даден обект. Тогава функцията *compare* връща 4, ако двете стойности съвпадат и са равни на „+“, 5 ако съвпадат и са равни на „-“ и 6, ако не съвпадат. Правилото за празните клетки също важи.

След това можем да започнем да натрупваме сумите в *sum_m_c*, *sum_n_c* и *sum_p_c*. За всяка двойка обекти се работи с всяка двойка критерии с номера съответно *i* и *j* (първият – от първата таблица, вторият – от втората). С помощта на *compare* се проверява знака между оценките на двата обекта по първия критерий. След това отново с *compare* се проверява знакът между оценките на двата обекта по втория критерий.

Ако двата знака съвпадат, но не са знак „=“, то се увеличава с единица броячът за принадлежност за двойката критерии в *sum_m_c[i][j]*. Ако знаците не съвпадат съответно се увеличава с единица броячът за непринадлежност сума в *sum_n_c[i][j]*. Ако пък двата знака съвпадат и са „=“, действието зависи от настройките, избрани при отваряне на входния файл. Изпълнява се един от следните варианти:

- При два знака равенство *sum_m_c[i][j]* и *sum_n_c[i][j]* получават по 0,5.
- При два знака равенство *sum_n_c[i][j]* се увеличава с единица
- При два знака равенство *sum_m_c[i][j]* се увеличава с единица
- При два знака равенство се увеличава *sum_p_c[i][j]*.

С цел да се ускори работата на програмата в тази ѝ част, резултатите от сравненията между двойки числа с *compare* се изчисляват предварително за всяка дадена двойка колони (обекти) от двете входни матрици. Те се съхраняват в масиви *x_values* (за сравненията на числа от *data1*) и *y_values* (за сравненията от *data2*), където позицията на даден резултат от сравнение се определя от номера на реда, на който се намират двете числа.

Тъй като проверките за всяка двойка колони *k1* и *k2* са независими, е възможно всяка двойка колони да се отдели в отделна нишка на процеса, в който се изпълнява приложението.

Това е полезно най-вече при системи с повече от едно процесорно ядро и ползволява всяко ядро да обработва отделна двойка колони паралелно с другите ядра. Това обаче налага някои структурни промени в програмния код, за да се предпазят стойностите на броячите *sum_m_c[i][j]*, *sum_n_c[i][j]* и *sum_p_c[i][j]* от сблъсъци между различните нишки. За целта е създаден класа *CalcThread*, наследник на *QThread*, в който са добавени полета *sum_m_c*, *sum_n_c* и *sum_p_c*, идентични по тип на тези в *MainWindow*. Така всяка нишка притежава самостоятелни броячи, а след края на работата на нишките, стойностите на техните броячи се сумират в *MainWindow*.

Приложението заделя толкова процесни нишки, колкото са наличните ядра на централния процесор (който може да бъде повече от един при някои конфигурации). Нишките се съхраняват в масив от указатели, като изчисленията за всяка следваща двойка колони се насочват последователно към следващата по ред нишка от масива.

Самите операции по сравнение и решаване кога и кой брояч да се увеличи са отделени в отделен клас, наречен *Calculations*, а по-точно в неговия метод *doCalculate*. За всяка двойка колони се създава отделен обект от този клас. Основният клас на главния прозорец *MainWindow* предава данните от входните две матрици посредством указатели към *data1* и *data2*. След като към обекта за изчисления се препратят указатели към необходимите данни, както и номерата на сравняваните колони от входните матрици, обектът се прехвърля към една от наличните процесни нишки. Резултатите от сравненията се съхраняват в броячите на съответната нишка.

За да не се достигне до загуба на данни приложението изчаква броят на обработените двойки колони да достигне общия брой колони. Едва когато този брой е достигнат се преминава към следващата стъпка.

След като всички обекти от класа *Calculations* приключат работа, стойностите на броячите, натрупани във всяка нишка, се сумират и резултатът се записва в главните броячи от *MainWindow.*

След като получим всички суми, трябва да се изчислят стойностите на μ и ν за всяка двойка критерии от първата и втората таблица. Ако N е броят на обектите, то това става по следния начин

за всеки два критерия с номер i и j (първият – от първата таблица, вторият – от втората):

$$Ro[i][j] = \frac{sum\_m[i][j]}{\frac{N(N-1)}{2}}$$

$$Sugma[i][j] = \frac{sum\_n[i][j]}{\frac{N(N-1)}{2}}$$

При завършване на работата връзката с COM сървъра на Excel се прекъсва.

4.3.Други използвани класове

След получаване на резултатите данните в тях трябва да бъдат изведени на екрана. За целта в главния прозорец са поставени един под друг обекти от клас QTableView, който позволява да се изгради таблица с допълнително описан модел. Тук е използван класът *VectorTableModel*, наследник на класа *QAbstractTableModel*. Той позволява от двумерния вектор данни да се четат само тези стойности, които са видими в таблицата.

Библиотеката Qt предоставя възможности за работа с двуизмерна графика посредством класовете *QGraphicsView* и *QGraphicsScene*. Прозорецът за изобразяване на графика е съставен с помощта на класа за прозорец *IFS_Triangle*, в който е вмъкнат обект от новия клас *IFS_GraphicsView*, който наследява *QGraphicsView*. Създаването на собствен клас-наследник за графичното платно позволява по-голяма гъвкавост при изрисуването на обектите.

Интуиционистки размитият триъгълник се изобразява с помощта на вградения клас *QGraphicsPolygonItem*, който е предназначен за чертане на многоъгълници. Точките могат да се изобразят като отделни окръжности, но тъй като се очаква броят им да бъде значително по-голям (което ще натовари значително паметта), вместо това те се рисуват директно на фона на графичното платно. Дублиращите се точки се премахват предварително с цел по-бързо действие. Самото изчертаване се

контролира от класа на централния прозорец, тъй като там се съхраняват входните и изходните данни.

Библиография

Споменати в текста публикации:

1. Atanassov, K., D. Mavrov, V. Atanassova. Intercriteria Decision Making: A New Approach for Multicriteria Decision Making, Based on Index Matrices and Intuitionistic Fuzzy Sets. Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 11, 2014, 1–8, ISBN: 978-83-61551-10-2.
2. Atanassova, V., L. Vardeva, E. Sotirova, L. Doukovska, Traversing and ranking of elements of an intuitionistic fuzzy set in the intuitionistic fuzzy interpretation triangle, Chapter, Novel Developments in Uncertainty Representation and Processing, Vol. 401, Advances in Intelligent Systems and Computing, 2016, 161-174.

Програмата е представена със следните конференции и публикации:

a) Mavrov, D. Software for InterCriteria Analysis: Implementation of the main algorithm, Notes on Intuitionistic Fuzzy Sets, Vol. 21, 2015, No. 2, 77-86.
b) Mavrov, D., I. Radeva, K. Atanassov, L. Doukovska, I. Kalaykov, InterCriteria Software Design: Graphic Interpretation within the Intuitionistic Fuzzy Triangle, Proceedings of the Fifth International Symposium on Business Modeling and Software Design, 2015, 279-283.
c) Mavrov, D. Software for InterCriteria Analysis: Working with the Results. Annual of “Informatics” Section Union of Scientists in Bulgaria Volume 8, 2015.