Универзитет у Београду

Електротехнички факултет



Конфигурабилни стохастички симулатор епидемија са паралелним извршавањем

Пројектни задатак

|  |  |
| --- | --- |
| Ментор: | Кандидат: |
| др Бранко Маровић | Стефан Милановић 2019/3098 |

Београд, Фебруар 2020.

Садржај

[Садржај 2](#_Toc31326363)

[1. Увод 3](#_Toc31326364)

[2. Дизајн и имплементација решења 4](#_Toc31326365)

[2.1. Стохастички модели епидемија 4](#_Toc31326366)

[2.2. Дизајн система 4](#_Toc31326367)

[2.3. Коришћене библиотеке 4](#_Toc31326368)

[2.4. Имплементација система 5](#_Toc31326369)

[2.4.1. Класа ConfigFileParser (ConfigFileParser.h, ConfigFileParser.cpp) 5](#_Toc31326370)

[2.4.2. Класа Configuration (Configuration.h, Configuration.cpp) 5](#_Toc31326371)

[2.4.3. Класа Simulator (Simulator.h, Simulator.cpp) 5](#_Toc31326372)

[2.4.4. Класа SimulationInfo (SimulationInfo.h, SimulationInfo.cpp) 6](#_Toc31326373)

[2.4.5. Структура RecordedData (SimulationInfo.h) 6](#_Toc31326374)

[2.4.6. Структура Event (SimulationInfo.h) 6](#_Toc31326375)

[3. Начин коришћења апликације 7](#_Toc31326376)

[3.1. Превођење изворног кода и формат конфигурационе датотеке 7](#_Toc31326377)

[3.1.1. Превођење изворног кода 7](#_Toc31326378)

[3.1.2. Формат конфигурационе датотеке 7](#_Toc31326379)

[3.2. Формат излазних датотека 8](#_Toc31326380)

[4. Пример анализе података 10](#_Toc31326381)

[4.1. Имплементација система 10](#_Toc31326382)

[4.2. Имплементација система 10](#_Toc31326383)

[5. Тачке унапређења и проширења апликације 11](#_Toc31326384)

[5.1. Тачке унапређења апликације 11](#_Toc31326385)

[5.2. Тачке проширења апликације 11](#_Toc31326386)

1. Увод

Овај документ представља извештај израде пројектног задатка на предмету Рачунарство у биомедицини (13М111RBM) у школској 2019/2020. години. Циљ пројектног задатка је реимплементација стохастичог симулатора епидемија који је написао Давор Остојић у оквиру свог мастер рада. У оквиру ове имплементације симулатора додата је подршка за конфигурацију почетних параметара, покретање и извршавање већег броја симулација истовремено, и подршка за форматирано исписивање резултата извршавања свих симулација. Поред описа техничких аспеката пројекта као што су објашњења имплементираних класа и формат излазних датотека, овај документ садржи и малу дискусију о стохастичким моделима епидемија и дискусију примене ансамбала на епидемије (уз представљање примера анализе коришћењем имплементираног симулатора).

Друго поглавље овог документа садржи поменуту дискусију о стохастичким моделима епидемија, главне одлуке приликом дизајна система, и опис имплементације решења. Ово поглавље истиче стохастичке моделе који су подржани од стране симулатора и наводи алтернативне моделе који се могу користити за симулације епидемија, али који тренутно нису подржани од стране симулатора. У оквиру овог поглавља се такође наводе коришћене библиотеке и опис имплементације уз навођење реализованих класа.

У трећем поглављу биће описан начин коришћења апликације, тачније опис поступка превођења изворног кода и формата конфигурационе датотеке који се користи за параметризовање свих симулација. Пошто не постоји експлицитна интеракција између корисника и апликације (интеракција се врши преко конфигурационе датотеке), акценат овог поглавља јесте објашњавање формата конфигурационе датотеке. Такође, у овом поглављу ће бити описани формат излазних датотека зависно од типа симулације.

У четвртом поглављу приказана су два примера анализе података добијених покретањем симулатора са одговарајућим почетним параметрима. За сваки пример биће наведени циљ анализе, почетна конфигурација симулатора (преглед конфигурационе датотеке пре покретања симулација), резултати симулација и дискусија анализе резултата.

У последњем поглављу овог документа наводе се могуће тачке унапређења и проширења апликације. Тачке унапређења апликације представљају функционалности које апликација већ поседује али се могу у одређеној мери дорадити, док су тачке проширења апликације корисне функционалности које не постоје у тренутној верзији апликације.

1. Дизајн и имплементација решења

Пре посматрања начина коришћења симулатора и начина анализе резултујућих датотека, неопходно је најпре изложити начин дизајна и имплементације решења.

Ово поглавље ће бити подељено у 4 потпоглавља. У првом делу овог поглавља биће направљена мала дискусија о стохастичким моделима епидемија. Након тога ће у оквиру другог потпоглавља бити изнете главне одлуке приликом дизајна система. У трећем потпоглављу биће истакнуте коришћене библиотеке и разлози за њихово коришћење. Коначно, у последњем потпоглављу биће детаљно описана имплементација симулатора уз навођење реализованих класа и њихових кратких описа.

* 1. Стохастички модели епидемија

Модели епидемија могу бити детерминистички или стохастички.

* 1. Дизајн система

У овом поглављу наводе се главне одлуке дизајна система:

1. **Стохастички модели епидемија** – решење користи стохастички моделе епидемија који су лакши за софтверску имплементацију и погодни за величине популација до неколико хиљада јединки. Главни проблем стохастичких модела се огледа у томе што је неопходно покренути велики број симулација и радити статистичку анализу резултата како би се донели адекватни закључци. Овај пројекат решава управо тај проблем; СИР, СЕИР, СЕИР-симп
2. **Имплементација у језику C++ --** имплементација решења је урађена у програмском језику C++ због подршке компајлера за библиотеку *OpenMP* која се користи приликом паралелизације кода;
3. **Интуитиван кориснички интерфејс –** конфигурациона датотека и излазне датотеке дизајниране су тако да буду читљиве и употребљиве без познавања тачних спецификација њихових формата;
4. **Једноставно коришћење –** како би корисник користио симулатор неопходно је једино да ажурира конфигурациону датотеку, а затим прочита резултате записане у излазним датотекама (како би након покретања симулација могао радити анализу резултата);
5. **Модуларан изворни код –** изворни код мора је модуларан како би се олакшале евентуалне модификације или проширења решења.
   1. Коришћене библиотеке

Приликом имплементације решења коришћене су две помоћне библиотеке:

1. **nlohmann/json –** ова библиотека нуди интерфејс за парсирање *.json* датотека и омогућава лако учитавање конфигурационе датотеке коју корисник ажурира;
2. **OpenMP –** апликативни кориснички интерфејс који подржава мултипроцесорско програмирање у системима са дељеном меморијом у програмским језицима *C, C++,* и *Fortran.* Ова библиотека се користи за паралелизацију кода који ради обраду свих симулација и исписује резултате у одговарајуће излазне датотеке (број симулација специфицира корисник у конфигурационој датотеци).
   1. Имплементација система

У оквиру овог потпоглавља биће изложене све реализоване класе и структуре уз кратак опис и објашњење улоге те класе у систему, као и повезаност .

* + 1. Класа ConfigFileParser (ConfigFileParser.h, ConfigFileParser.cpp)

Ова класа служи за парсирање и учитавање улазне конфигурационе датотеке чије је име специфицирано у датотеци *main.cpp.* Подразумевани назив конфигурационе датотеке је *config.conf* и неопходно је да се она налази у истом директоријуму као и извршни фајл симулатора.

Метода *parse()* ове класе на основу очекиваног формата конфигурационе датотеке учитава све неопходне податке и параметре и прослеђује их објекту класе *Configuration.* Ова метода не врши никакву обраду над учитаним подацима.

* + 1. Класа Configuration (Configuration.h, Configuration.cpp)

У оквиру ове класе се чувају подаци који су учитани од стране објекта класе *ConfigFileParser.* Класа се састоји од већег броја *getter* и *setter* функција које се користе у осталим класама овог система.

* + 1. Класа Simulator (Simulator.h, Simulator.cpp)

Ова класа служи за складиштење информација о свим симулацијама и за покретање главне обраде система.

Метода *simulate()* ове класе најпре креира онолико симулација колико је специфицирано у конфигурационој датотеци (свака појединачна симулација представљена је једним објектом класе *SimulationInfo* која је описана у ставци *2.4.4.*), а затим и креира директоријум у којем ће излазне датотеке са резултатима бити сачуване. Након ових припремних корака врши се расподела свих симулација на одговарајући број нити (који је такође специфициран у конфигурационој датотеци), при чему нити обрађују сваку симулацију на идентичан начин почев од времена 0 (време почетка симулације) који се састоји од наредних корака:

1. Ажурирање вероватноћа свих подржаних елементарних догађаја (описаних у поглављу *2.1.*), при чему се игноришу догађаји који не одговарају типу симулације;
2. Налажење наредног тренутка у времену када се дешава неки од посматраних елементарних догађаја;
3. Избор елементарног догађаја који се у претходно израчунатом тренутку дешава (ово се рачуна на основу претходно израчунатих вероватноћа елементарних догађаја у овој итерацији);
4. Провера испуњености услова догађаја који се још нису догодили (подржани догађаји су описани у поглављу *2.1.*);
5. Памћење стања итерације за каснији испис у излазну датотеку.

Ова обрада се понавља циклично или до нестанка инфицираних јединки или до задатог временског тренутка (задаје се у конфигурационој датотеци).

Свака нит, када заврши са обрадом симулације, врши испис резултата те симулације у излазну датотеку. Назив излазне датотеке садржи аутоматски додељени идентификатор посматране симулације. Нит завршава са својим извршавањем уколико су све симулације које су њој расподељене обрађене, а у супротном понавља овај поступак за наредну необрађену симулацију.

Постоји имплицитна синхронизациона тачка за све нити у оквиру методе *simulate()*, тако да се обрада свих симулација завршава у оквиру ове методе.

* + 1. Класа SimulationInfo (SimulationInfo.h, SimulationInfo.cpp)

Ова класа представља једну симулацију у оквиру главне обраде програма описане у претходној ставци. Објекти ове класе имају јединствени идентификатор и конкретне вредности свих популација и параметара (у оквиру конфигурационе датотеке задају се опсези вредности популација и параметара, а затим се приликом креирања објеката класе *SimulationInfo* на псеудослучајан начин одређују конкретне вредности ових података).

У оквиру ове класе се памти листа стања популација и догађаја који се ажурирају у току симулације, а такође се памте и додатни подаци као што су број рођених јединки, број мртвих јединки, итд.

* + 1. Структура RecordedData (SimulationInfo.h)

Ова структура представља стање популација и додатних података у одређеном временском тренутку у току симулације. Памти се тренутак када је стање настало (енгл. *timestamp*), вредности популација у том тренутку и вредности додатних података у том тренутку.

Користи се за исписивање свих стања симулације у излазни фајл.

* + 1. Структура Event (SimulationInfo.h)

Структура *Event* се користи за памћење информација о догађајима који су присутни у симулацији. У оквиру ове класе постоји индикатор о томе да ли је догађај остварен или не, као и назив догађаја.

Тачка кретања програма налази се у датотеци *main.cpp.* Функција *main* креира један објекат класе *Configuration* који помоћу интерног објекта класе *ConfigFileParser* парсира конфигурациону датотеку. Овај објекат се затим прослеђује објекту класе *Simulator* помоћу чега се обезбеђује приступ параметрима од стране свих паралелно извршаваних симулација. Након овога позивом методе *simulate()* објекта класе *Simulator* извршава се главна обрада пројекта описана у ставци *2.4.3.*

1. Начин коришћења апликације

Пошто корисник не интерагује директно са симулатором, већ путем конфигурационе датотеке, у овом поглављу детаљно ће бити описан њен формат, као и формат излазних датотека које се креирају након покретања симулатора. Уједно ће бити истакнуте напомене за превођење изворног кода симулатора уз навођење развојног окружења како би корисник могао самостално преводити пројекат.

* 1. Превођење изворног кода и формат конфигурационе датотеке
     1. Превођење изворног кода

Превођење изворног кода се наводи као засебна ставка овог документа зато што је због коришћене **OpenMP** библиотеке неопходно преводиоцу навести додатне параметре за обраду изворног кода.

Окружење које је коришћено за превођење је *Microsoft Visual Studio Enterprise 2017 – Version 15.8.6.* Поред подразумеваних параметара које *Visual Studio* користи приликом превођења *C++* пројеката, неопходно је експлицитно захтевати подршку за *OpenMP*навођењем директиве ***/openmp,*** а у командној линији додати опцију ***/Zc:twoPhase-***.

* + 1. Формат конфигурационе датотеке

Конфигурациона датотека је организована као *.json* датотека и комуникација између корисника и самог симулатора се одвија преко ове датотеке. Корисник не треба да мења формат датотеке, већ само вредности приложених параметара. Након ажурирања ове датотеке, корисник простим покретањем извршног фајла симулатора извршава жељену симулацију.

У остатку овог поглавља биће детаљно објашњен формат конфигурационе датотеке уз навођење значења сваког поља и типа вредности који симулатор очекује за одговарајућа проста поља.

Конфигурациона датотека се на првом нивоу дели на 4 секције:

1. ***general*** *–* у оквиру ове секције датотеке наводе се општи параметри симулатора. Ови параметри су:
   1. *SimulationType –* коришћени стохастички модел епидемија приликом извршавања симулација. Промена овог параметра утиче на промену коришћених елементарних догађаја и процеса (описаних у потпоглављу *2.1.*) и промену формата излазних датотека (у излазним датотекама се приказују само подаци релевантни за изабрани тип симулације). Тип овог параметра је ниска која мора имати једну од наредних вредности: “SIR”, “SEIR”, “SEIR\_simplified”;
   2. *SimulationDuration –* поље *maximum\_duration(days)* из овог параметра одређује тренутак до којег се прати симулација. Тип овог параметра је број. Уколико је његова вредност 0, симулација се извршава док број инфицираних (и изложених (енгл. *еxposed*), у случају *SEIR* модела) не износи 0. У супротном, симулација се извршава до прослеђеног тренутка у времену.
   3. *NumberOfSimulations –* ово поље одређује укупан број симулација који ће се извршити паралелно. На крају рада симулатора у фолдеру *output\_files* налазиће се оволико датотека које садрже резултате симулације. Тип овог параметра је целобројна вредност.
   4. *NumberOfThreads –* ово поље одређује број нити коришћених приликом рада симулатора. Корисник може подешавати овај параметар сходно могућностима свог рачунара. Тип овог параметра је целобројна вредност.
2. ***populations*** *–* ова секција датотеке омогућава задавање опсега вредности за све групе популација (вредности параметара популације изложених се игноришу уколико је за тип симулације одабран *SIR* модел). Опсези се конфигуришу променом  *lower\_bound* и *upper\_bound* вредности сваке од приложених популација – *Susceptible, Exposed, Infected,* и *Recovered.* Очекивани тип ових параметара је целобројна вредност.
3. ***parameters*** *–* ова секција датотеке омогућава задавање опсега вредности за све параметре свих симулација. Опсези се конфигуришу променом  *lower\_bound* и *upper\_bound* вредности сваког од приложених параметара – *Mortality Rate, Infected Mortality Rate, Recovery Rate, Incubation Period,* и *Infection Rate.* Очекивани тип ових параметара је децимални број чија је вредност између 0 и 1;
4. ***events –*** ова секција датотеке омогућава конфигурисање два подржана догађаја од стране симулатора *–* *Vaccination* и *Revaccination* (описаних у потпоглављу *2.1*)*.*
   1. *Vaccination –* за догађај вакцинације се везује временски тренутак који се генерише на псеудослучајан начин и узима вредност из опсега *timestamp\_lower\_bound* и *timestamp\_upper\_bound*, као и ефикасност вакцинације одређена пољем *vaccination\_efficiency.* Тип ових параметара је децимални број, при чему за поље *vaccination\_efficiency* вредност мора бити између 0 и 1. Тип поља *used* је вредност *true* или *false* и одређује да ли се догађај уопште разматра приликом симулација;
   2. *Revaccination –* за догађај ревакцинације везује се параметарефикасности ревакцинације *revaccination\_efficiency.* Тип овог параметра је децимални број чија је вредност између 0 и 1. Тип поља *used* користи се аналогно као и код догађаја вакцинације.
   3. Формат излазних датотека

У овом потпоглављу представљени је формат излазних датотека. Реални примери који су генерисани покретањем симулатора се налазе у директоријуму *example\_output\_files* који је приложен уз овај документ, при чему сваки пример одговара једном типу симулације.

На почетку свих излазних датотека најпре се наводи назив симулације који укључује њен јединствени идентификатор. Након овога све излазне датотеке деле се у 6 секција:

1. **Опис општих информација** – кратак преглед општих информација постављених у коришћеној конфигурацији када је посматрана симулација покретана
2. **Иницијалне популације** – преглед иницијалног стања свих популација посматране симулације
3. **Вредност параметара** – преглед вредности параметара посматране симулације (параметри се не мењају у току извршавања симулације);
4. **Активни догађаји** – списак догађаја који су узети у обзир приликом извршавања симулације (догађај је активан уколико је вредност поља *used* у конфигурацији постављена на *true* пре покретања симулатора);
5. **Извештај симулације** – најдужа секција излазног фајла која садржи табелу са релевантним информацијама за тип посматране симулације. У овој табели креира се ред за сваки тренутак у времену када се догодио елементарни догађај у току симулације. Број и вредност колона табела зависе од типа симулације и представљени су у наставку:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time** | **Susceptible** | **Infected** | **Recovered** | **Total population** | **Births** | **Deaths – Susceptible** | **Deaths – Infected** | **Deaths – Recovered** | **Deaths – Due to Infection** | **Deaths - Total** |

**Табела 1. Преглед колона табеле извештаја симулације – SIR модел**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Time** | **Susceptible** | **Exposed** | **Infected** | **Recovered** | **Total population** | **Births** | **Deaths – Susceptible** | **Deaths – Infected** | **Deaths – Recovered** | **Deaths – Due to Infection** | **Deaths - Total** |

**Табела 2. Преглед колона табеле извештаја симулације – SЕIR модел**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time** | **Susceptible** | **Exposed** | **Infected** | **Recovered** | **Total population** |

**Табела 3. Преглед колона табеле извештаја симулације – SЕIR\_simplified модел**

1. **Преглед симулације –** садржи скраћену верзију табеле из пете секције излазне датотеке која садржи само први и последњи ред. Ова секција је од користи када се желе посматрати само почетно и крајње стање симулације.
2. Пример анализе података
   1. Имплементација система

Верификација је одрађена ручно упоређивањем примарног скупа података са секундарним скупом података. Понављањем инспекције и верификацијом успешности написаних скрипти за креирање .*gml* фајлова потврђује се да су подаци очишћени и да се сада налазе у адекватном формату за формирање тражених мрежа.

* 1. Имплементација система

Верификација је одрађена ручно упоређивањем примарног скупа података са секундарним скупом података. Понављањем инспекције и верификацијом успешности написаних скрипти за креирање .*gml* фајлова потврђује се да су подаци очишћени и да се сада налазе у адекватном формату за формирање тражених мрежа.

1. Тачке унапређења и проширења апликације

У последњем поглављу овог документа дискутују се могуће тачке унапређења и тачке проширења система.

* 1. Тачке унапређења апликације

Тачке унапређења апликације представљају функционалности које апликација већ поседује али се могу у одређеној мери дорадити. За овај систем предлажу се наредна унапређења:

1. **Подршка за опоравак у случају грешке (енгл. *error handling*) –** тренутна имплементација симулатора не садржи подсистем за проверу некоректно унетих података од стране корисника у конфигурационом фајлу (нпр. негативан број симулација, прослеђена је невалидна *.json* датотека, опсег код којег је вредност поља *lower\_bound* већа од вредности поља *upper\_bound);*
2. **Побољшање флексибилности решења –** тренутна имплементација решења има подршку за тачно 3 типа симулација (*SIR, SEIR, SEIR\_simplified*), тачно 2 догађаја (вакцинација и ревакцинација) и за тачно 8 процеса (рађање нове јединке, смрт подложне јединке, зараза, смрт оболеле јединке, смрт опорављене јединке, инфекција, смрт јединке због инфицираности, и опоравак). Решење се може направити флексибилнијим пружањем подршке за дефинисање догађаја и процеса од стране корисника;
3. **Нови формати излазних датотека –** резултати симулатора се могу корисницима пружити и у другим форматима који би били кориснији за другачији начин обраде резултата покренутих симулација. Тренутна имплементација симулатора подржава само један формат излазних датотека.
   1. Тачке проширења апликације

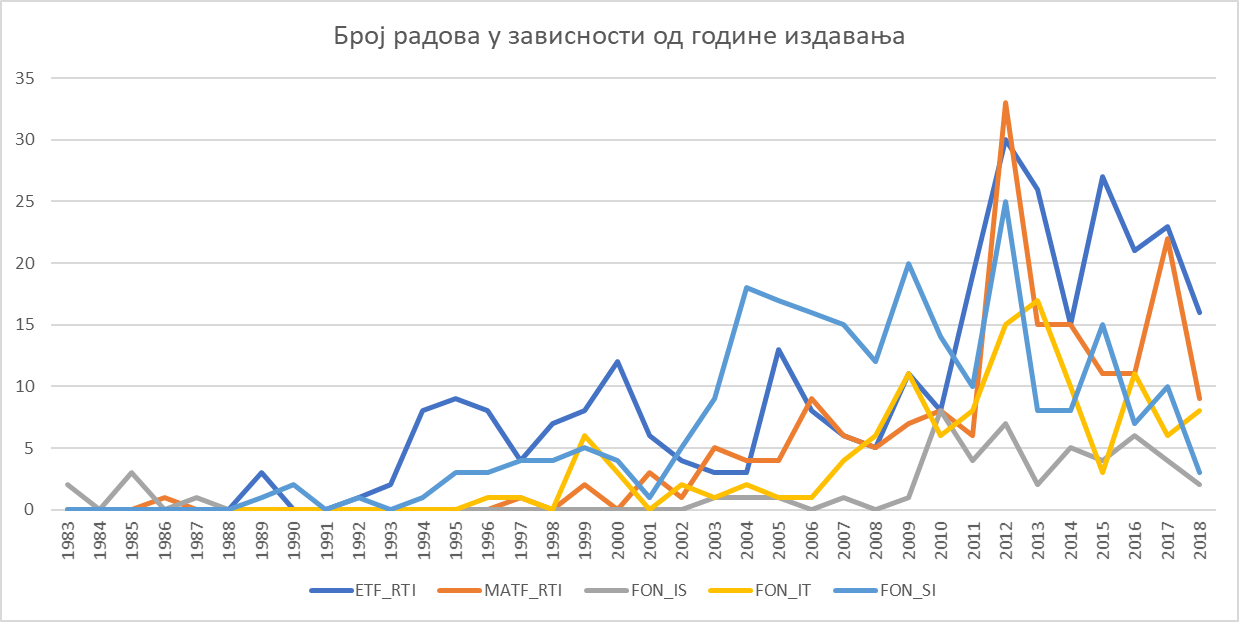
Тачке проширења апликације су корисне функционалности које не постоје у тренутној верзији апликације. За овај систем предлажу се наредна проширења:

1. **Графички кориснички интерфејс –** систем се може проширити пребацивањем података из конфигурационог фајла у графички кориснички интерфејс помоћу којег корисник може мењати параметре симулатора, а затим и покренути све симулације;
2. **Додавање метода за статистичко праћење епидемија –** систем се може унапредити имплементацијом метода за статистичко праћење епидемија као што је *Ensemble Optimal Statistical Interpolation (EnOSI)* метод или *Еnsemble Kalman filter (EnKF).*

Најпродуктивнији научници се налазе рангирањем укупног броја радова сваког аутора, укључујући и радове које су аутори сами писали или на којима су сарађивали са коауторима који се не разматрају приликом креирања мреже.

Табела десет најпродуктивнијих научника, као и њихов укупан број радова и идентификатор катедре и факултета којем припадају, приказана је у наставку:

|  |  |  |
| --- | --- | --- |
| ***Author*** | ***Department*** | ***Number of papers*** |
| Владан Девеџиж | FON\_SI | 137 |
| Јелена Јовановић | FON\_SI | 106 |
| Сања Вранеш | ETF\_RTI | 68 |
| Душан Старчевић | FON\_IT | 58 |
| Мило Томашевић | ETF\_RTI | 46 |
| Мирослав Миновић | FON\_IT | 46 |
| Милош Миловановић | FON\_IT | 39 |
| Предраг Јаничић | MATF\_RTI | 37 |
| Бошко Николић | ETF\_RTI | 34 |
| Зоран Јовановић | ETF\_RTI | 34 |

На наредној слици је проказан график зависности броја радова од године издавања за сваку од катедри. Може се приметити да су све катедре 2012. године имале веома велики број радова, а чак три од њих тада имају максимум издатих радова.

У табели је за сваку катедру излистана година када је она публиковала највише радова и број тих радова.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ETF\_RTI | MATF\_RTI | FON\_IS | FON\_IT | FON\_SI |
| ***Maximum*** | 30 | 33 | 8 | 17 | 25 |
| ***Year*** | 2012 | 2012 | 2010 | 2013 | 2012 |