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](#_Toc31403118)

[1. Увод 3](#_Toc31403119)

[2. Дизајн и имплементација решења 4](#_Toc31403120)

[2.1. Стохастички модели епидемија 4](#_Toc31403121)

[2.2. Дизајн система 5](#_Toc31403122)

[2.3. Коришћене библиотеке 6](#_Toc31403123)

[2.4. Имплементација система 6](#_Toc31403124)

[2.4.1. Класа ConfigFileParser (ConfigFileParser.h, ConfigFileParser.cpp) 6](#_Toc31403125)

[2.4.2. Класа Configuration (Configuration.h, Configuration.cpp) 6](#_Toc31403126)

[2.4.3. Класа Simulator (Simulator.h, Simulator.cpp) 6](#_Toc31403127)

[2.4.4. Класа SimulationInfo (SimulationInfo.h, SimulationInfo.cpp) 7](#_Toc31403128)

[2.4.5. Структура RecordedData (SimulationInfo.h) 7](#_Toc31403129)

[2.4.6. Структура Event (SimulationInfo.h) 7](#_Toc31403130)

[3. Начин коришћења апликације 9](#_Toc31403131)

[3.1. Превођење изворног кода и формат конфигурационе датотеке 9](#_Toc31403132)

[3.1.1. Превођење изворног кода 9](#_Toc31403133)

[3.1.2. Формат конфигурационе датотеке 9](#_Toc31403134)

[3.2. Формат излазних датотека 11](#_Toc31403135)

[3.2.1. Формат .txt фајлова 11](#_Toc31403136)

[3.2.2. Формат .csv фајлова 11](#_Toc31403137)

[3.3. Покретање симулатора 12](#_Toc31403138)

[4. Пример анализе података 13](#_Toc31403139)

[4.1. Налажење утицајности појединачних параметара на трајање епидемије 13](#_Toc31403140)

[4.1.1. Варијабилна стопа смртности 13](#_Toc31403141)

[4.1.2. Варијабилна стопа смртности инфицираних јединки 14](#_Toc31403142)

[4.1.3. Варијабилна стопа опоравка 15](#_Toc31403143)

[4.1.4. Варијабилна стопа инфективности 16](#_Toc31403144)

[4.1.5. Закључак 16](#_Toc31403145)

[5. Тачке унапређења и проширења апликације 18](#_Toc31403146)

[5.1. Тачке унапређења 18](#_Toc31403147)

[5.2. Тачке проширења 18](#_Toc31403148)

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. Поред реимплементације симулатора додата је подршка за конфигурацију опсега 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 формата излазних датотека, овај документ садржи и малу дискусију о стохастичким моделима епидемија.

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 које су неопходне за омогућавање рада система.

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 тумачења 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 и назначено који модели су подржани од стране имплементираног симулатора. 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. Детерминистички модели епидемија се користе када се посматрају велике популације и они за исту вредност улазних параметара увек дају идентичне излазне параметре. Поред овога, детерминистички модели нису погодни за софтверску имплементацију. Из овог разлога овај симулатор подржава стохастичке моделе епидемија.

Постоје стохастичке формулације разних детерминистичких модела епидемија. У наставку се наводи пар примера ових модела:

1. *SIR* (*Susceptible, Infectious, Recovered*)модел – популација се дели на 3 групе: јединке које су подложне болести спадају у групу *Susceptible,* јединке које су заразне спадају у групу *Infectious,* док су опорављене (или мртве) јединке смештене у *Recovered* групи. Овај модел је користан за моделовање инфективних болести које се преносе од човека до човека;
2. *SIS* (*Susceptible, Infectious, Susceptible*) модел – у оквиру овог модела јединке могу бити или у групи подложних болести, или у групи заразних (не постоји стање имунитета). Овај модел је користан за моделовање инфекција код којих се не ствара трајни имунитет јединки;
3. *MSIR* (*Maternally-derived immunity, Susceptible, Infectious, Recovered*) модел – увођењем нове групе *Maternally-derived immunity* омогућава класификацију јединки које су иницијално имуне од болести, али касније постају подложне њој;
4. *SEIR* (*Susceptible, Exposed, Infectious, Recovered*) модел – овај модел додаје нову групу *Exposed* у којој се налазе јединке које су оболеле од посматране болести, али још нису заразне и не могу ту болест преносити даље.

Постоје подваријанте поменутих модела као што су *SEIS* и *MSEIR* модели, али ови модели неће бити разматрани у овом документу. Такође, постоје алтернативни стохастички модели епидемија као што je *Erlang SEIR* модел који пружа реалистичније рачунање транзиција јединки између група у *SEIR* моделу него што то ради основни *SEIR* модел.

Стохастички модели у овом симулатору спадају у категорију *Continuous Time Markov Chain* (*CTMC*) модела код којих су вероватноће процеса дискретне вредности, а експоненцијално расподељено време континуално.

У оквиру овог симулатора подржани су наредни стохастички модели епидемија:

1. *SIR* (*Susceptible, Infectious, Recovered*) модел;
2. *SEIR* (*Susceptible, Exposed, Infectious, Recovered*) модел;
3. *SEIR\_simplified* (*Susceptible, Exposed, Infectious, Recovered*) модел – овај модел је заснован на специфичном моделу за анализу епидемије малих богиња у Абакаликију, Нигерија 1967. године који је користио Давор Остојић у свом раду. У овом моделу процеси рађања и умирања су избачени ради једноставности и због недостатака одговарајућих информација за проверу валидности резултата симулације. Такође, овај модел не садржи догађаје.

Тренутна имплементација симулатора подржава наредне елементарне догађаје:

1. *рађање;*
2. *зараза без могућности ширења болести;*
3. *зараза уз могућност ширења болести;*
4. *лечење јединке;*
5. *смрт подложне јединке;*
6. *смрт заражене јединке;*
7. *смрт заражене јединке* ***због леталности болести****;*
8. *смрт излечене јединке.*

Наредни параметри су подржани у тренутној имплементацији симулатора:

1. *стопа смртности*
2. *стопа смртности заражене јединке*
3. *стопа опоравка*
4. *период инкубације болести*
5. *стопа заразности*

Поред овога, симулатор подржава наредне догађаје:

1. *Вакцинација* – овај догађај се испуњава само једном у току симулације уколико је активиран. Услов за испуњавање овог догађаја је једино да симулација дође до конфигурисаног тренутка у времену. Уколико је овај услов испуњен, помоћу конфигурабилног фактора ефикасности вакцинације одређује се број подложних јединки који одмах прелази у групу излечених;
2. *Ревакцинација* – овај догађај се испуњава само једном у току симулације уколико је активиран. Услов за испуњавање овог догађаја јесте да се догађај вакцинације већ једном извршио, а да је број заражених јединки већи од 3/10 укупне популације. Уколико је овај услов испуњен, помоћу конфигурабилног фактора ефикасности ревакцинације одређује се број подложних јединки који одмах прелази у групу излечених.
   1. Дизајн система

У овом поглављу наводе се главне одлуке дизајна система:

1. **Подршка за стохастичке моделе епидемија** – решење подржава стохастичке 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;
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 (наведених 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.

Симулатор додатно исписује агрегиране податке из симулација у засебну датотеку *output\_simulations\_all*.*csv.*

* + 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 памти се 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.

Ова класа такође садржи и методе за исписивање у излазну датотеку у одговарајућем прослеђеном формату из објекта класе *Configuration.*

* + 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.

Сва стања у којима се симулација нађе у току извршавања се бележе и касније користе за исписивање у излазну датотеку.

* + 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 у тренутној верзији симулатора: “*SIR*”, “*SEIR*”, “*SEIR\_simplified*”;
   2. *OutputType –* жељени формат излазних фајлова. Тип овог предмета је ниска која мора имати једну од наредних вредности у тренутној верзији симулатора: “*txt*”, “*csv*”;
   3. *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.
   4. *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.
   5. *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.

* + 1. Формат .txt фајлова

На почетку свих излазних *.txt* датотека најпре се наводи назив симулације који укључује њен јединствени идентификатор. Након овога се излазна датотека дели 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 3.2.1.1. 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 3.2.1.2. Преглед колона табеле извештаја симулације – SЕIR модел**

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

**Табела 3.2.1.3 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.
   * 1. Формат .csv фајлова

Излазне датотеке овог формата прате стандардизовани *Comma Separated Values* формат у којем се у првом реду наводе називи колона, а у наредним редовима одговарајући подаци одвојени зарезима. Овај формат је користан за учитавање у Excel након чега се може радити обрада података из датотеке.

* 1. Покретање симулатора

Једино што је неопходно од корисника јесте да постави приложени извршни фајл *CSBMProject.exe* и конфигурационе датотеке *config.conf* у исти директоријум. Након овога, простим покретањем извршног фајла врши се парсирање конфигурационе датотеке и обрада симулација. На крају рада симулатора, аутоматски се у оквиру директоријума у којем се извршни фајл налази креира нови директоријум са називом *output\_files* у којем се налазе сви резултати покренутих симулација.

1. Пример анализе података

У оквиру овог поглавља дају се два примера обраде добијених података из симулатора. За сваки пример биће наведени циљ анализе, почетна конфигурација симулатора (преглед конфигурационе датотеке пре покретања симулација), резултати симулација и дискусија анализе резултата.

* 1. Налажење утицајности појединачних параметара на трајање епидемије

Циљ ове анализе је налажење параметра који највише утиче на трајање епидемије. Анализа ће бити извршена тако што се фиксирају почетна стања популација и сви параметри осим посматраног параметра. Фиксиране вредности параметара задају се опсегом код којег су доња и горња граница једнаке и њихове вредности ће бити једнаке просецима . За издвојени параметар.

За ову анализу користиће се стохастички *SIR* модел. Самим тим, неће бити посматран параметар *Incubation Period* јер се он не користи у процесима који су присутни у овом моделу.

За сваки посматрани параметар биће покренуто 5000 симулација. Посматрају се јединке код којих је 100 подложно инфекцији, а 1 јединка је иницијално инфицирана. Подразумеване вредности за параметре (када они нису варијабилни) су одређене на основу примера замишљене болести из Даворовог рада:

|  |  |  |  |
| --- | --- | --- | --- |
| **Mortality Rate** | **Infected Mortality Rate** | **Recovery Rate** | **Infection Rate** |
| 0.01 | 0.015 | 0.1 | 0.5 |

**Табела 4.1.1. Табела подразумеваних вредности параметара**

Резултати ових симулација су приказани на наредним графицима, при чему сваки график одговара једном од посматраних параметара:

* + 1. Варијабилна стопа смртности

Графикон на којем се приказује резултат варирања стопе смртности између вредности од 0 до 1 налази се у наставку:

**График 1. Резултат 5000 симулација са променљивом вредношћу *Mortality Rate* параметра**

Може се уочити да се појављује експоненцијална расподела трајања епидемије. Приликом тумачења резултата овог графикона можемо игнорисати све симулације чија је вредност стопе смртности изнад оквирно 0.4, зато што у тим случајевима читава популација јако брзо умире (невезано за стање у којем се јединка налази).

Посматрајући графикон можемо приметити да је расподела доста равномерна и да се прати тренд који наводи на закључак да се смањењем стопе смртности благо повећава просечна дужина трајања епидемије. Такође је битно уочити да се јавља велики број симулација у којима је дужина трајања епидемије била већа од 100 јединица времена, док пар симулација приказује епидемије чија је дужина трајања већа од 150 јединица времена.

* + 1. Варијабилна стопа смртности заражених јединки

Графикон на којем се приказује резултат варирања стопе смртности заражених јединки између вредности од 0 до 1 налази се у наставку:

**График 2. Резултат 5000 симулација са променљивом вредношћу *Infected Mortality Rate* параметра**

Као и на претходном графикону, и овде се може уочити појављивање експоненцијалне расподеле дужине трајања епидемије, само што је у овом случају експоненцијална функција стрмија него претходна. На овом графикону не треба игнорисати вредности параметра веће од 0.4 зато што је могуће да је анализирана болест која је изузетно летална.

Стрмија експоненцијална функција у овом случају означава да овај параметар драстично утиче на трајање епидемије једино када он тежи својим екстермним вредностима – 0 као минимуму, и 1 као максимуму. Ако посматрамо где се тачке налазе на графикону ако гледамо вредности параметра између 0.2 и 0.6, уочићемо да дужина трајања епидемија не прелази 15 јединица времена, док је углавном концентрисана између вредности 0 и 10.

Треба уочити да међу овим симулацијама постоји само једна симулација чија је дужина трајања епидемије износила више од 50 јединица времена, док су све остале симулације имале углавном доста краће трајање епидемије.

* + 1. Варијабилна стопа опоравка

Графикон на којем се приказује резултат варирања стопе опоравка између вредности од 0 до 1 налази се у наставку:

**График 3. Резултат 5000 симулација са променљивом вредношћу *Recovery Rate* параметра**

Као и претходна два графикона, и графикон овог параметра прати експоненцијалну расподелу. Разлика између овог графикона и претходних огледа се у томе што су чворови доста равномерније распоређени, али се може уочити да се смањењем вредности стопе опоравка повећава дужина трајања епидемије. Ово је посебно видљиво за вредности параметра између 0 и 0.3.

За разлику од графика из поглавља *4.1.1.* овде се не јавља велика фрагментација – као пример, посматрајмо дужину трајања епидемије 120 на оси *Epidemic duration* оба графикона. Можемо уочити да се на графикону варијабилне стопе смртности јављају симулације са овом дужином трајања епидемије код којих је стопа смртности близу 0.2, али постоје и симулације са истом дужином трајања код којих је стопа смртности близу 0.

На овом графику те појаве нема – дужину трајања епидемије 120 достижу једино симулације код којих је стопа опоравка мања од 0.1.

* + 1. Варијабилна стопа заразности

Графикон на којем се приказује резултат варирања стопе смртности инфицираних јединки између вредности од 0 до 1 налази се у наставку:

**График 4. Резултат 5000 симулација са променљивом вредношћу *Infection Rate* параметра**

Одмах примећујемо да се овај графикон доста разликује од претходних. Најпре, графикон не прати експоненцијалну расподелу, већ широку логаритамску расподелу. Након тога може се уочити да је, за разлику од свих осталих графикона, на овом графикону општа расподела чворова доста равномернија. Чворови нису нагомилани око кратких дужина трајања епидемија, већ су скоро равномерно распрострањени у опсегу од 0 до око 35 јединица времена.

Ово се може објаснити тиме што се повећањем стопе инфективности далеко повећава и вероватноћа да се епидемије неће одмах завршити, већ да ће се елементарни догађај инфекције појавити у току симулације. Ова особина доводи до тога да се просечно трајање епидемије помера са једне тачке на *x*-оси на читав опсег тачака.

Исто се може приметити да максимална дужина трајања епидемије у овим случајевима није велика, штавише мања је у односу на графиконе из поглавља *4.1.1.* и *4.1.3.* Ово се може објаснити тиме што велика вредност стопе инфективности доводи до тога да се читава популација брзо зарази, па се након тога само јављају смрти или опоравци до нестанка епидемије јер се не јавља баланс између броја подложних и броја инфицираних.

* + 1. Закључак

Закључак ове анализе јесте да се изолацијом појединачних параметара заиста могу приметити одређене правилности. Битно је напоменути да су покретане симулације пратиле почетне претпоставке описане у поглављу *4.1.* и да је могуће њиховим променама доста утицати на резултате анализе, али да се може очекивати да ће и са другим популацијама резултат бити сличан (само што би тада просечно трајање епидемије било дуже јер би било неопходно већи број јединки у просеку излечити).

Уз ову кратку анализу може се отворити дискусија о томе да ли је могуће на основу ових резултата одабрати најутицајнији параметар симулација.

Посматрајући графиконе добијене приликом анализе могу се издвојити два кандидат параметра – **стопа опоравка** и **стопа инфективности**. Стопа опоравка је кандидат за најутицанијији параметар зато што се са највећом сигурношћу може одредити њен утицај на трајање епидемије у односу на остале параметре. С друге стране, стопа инфективности се може сматрати кандидатом зато што је трајање епидемије највише нестабилно када се овај параметар варира.

Свакако, неопходно је детаљније тестирати и конфигурисати параметре и почетна стања симулација како би се потврдило да ли је овај података могуће одредити. Ово укључује промене иницијалне популације, број покренутих симулација, и стохастичког модела коришћеног за симулирање.

1. Тачке унапређења и проширења апликације

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

* 1. Тачке унапређења

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. **Уклањање ограничења система –** тренутна имплементација решења има ограничења у циљу побољшања перформанси као што су коришћење старије верзије *OpenMP* библиотеке која нема подршку за креирање послова (ово успорава извршавање целог програма у случају да су све нити осим једне завршиле своје симулације. Тада ће оне упослено чекати да та нит заврши са својим послом) и ограничење на максимално трајање симулације од 2 године;
4. **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 два формата излазних датотека (*.txt* и *.csv*).
   1. Тачке проширења

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 помоћу ансамбала –** систем се може унапредити имплементацијом метода за статистичко праћење епидемија као што је *Ensemble Optimal Statistical Interpolation (EnOSI)* метод или *Еnsemble Kalman filter (EnKF).*