DISKRETNA SIMULACIJA REDOVA ČEKANJA

Anđelić, Dominik

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Diplomski studij (Menadžment)

Dominik Anđelić

DISKRETNA SIMULACIJA REDOVA ČEKANJA

Diplomski rad

Osijek, 2022

Sveučilište Josipa Jurja Strossmayera u Osijeku
Ekonomski fakultet u Osijeku
Diplomski studij (Menadžment)

Dominik Anđelić

DISKRETNA SIMULACIJA REDOVA ČEKANJA

Diplomski rad

Kolegij: Poslovne simulacije

JMBAG: 0010221355
e-mail: dandelic@efos.hr

Mentor: doc. dr. sc. Domagoj Ševerdija

Osijek, 2022

Josip Juraj Strossmayer University of Osijek
Faculty of Economics in Osijek
Graduate Study (Management)

Dominik Anđelić

DISCRETE SIMULATION OF QUEUES

Graduate paper

Osijek, 2022

IZJAVA

O AKADEMSKOJ ČESTITOSTI, PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA, SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA

1. Kojom izjavljujem i svojim potpisom potvrđujem da je diplomski (navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. (cc) BY-NC-SA
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

Ime i prezime studenta/studentice: Dominik Anđelić

JMBAG: 0010221355

OIB: 775 282 48512

e-mail za kontakt: dominikandelic98@gmail.com

Naziv studija: Diplomski sveučilišni studij Poslovna ekonomija; smjer Menadžment

Naslov rada: Diskretna simulacija redova čekanja

Mentor/mentorica diplomskog rada: doc. dr. sc. Domagoj Ševerdija

U Osijeku, 14. lipnja 2022. godine

Potpis Dominik Anđelić

Diskretna simulacija redova čekanja

SAŽETAK

Rad prvotno prolazi kroz osnovnu teorijsku podlogu koja će pomoći pri razumijevanju ostatka rada. Nakon toga govori se o prijašnjim istraživanjima na temelju diskretnih simulacija redova čekanja. Iduće o čemu se govori jest detaljnija teoretska objašnjenja, mane i nedostaci diskretnih simulacija. Također, govori se o teoriji redova čime se bavi praktični dio ovog rada. U praktičnom primjeru pomoću Pythona, modeliran je sustav redova čekanja na primjeru jedne poslovne banke. Izvršena je simulacija koja se sastojala od 500 iteracija i testirana je učinkovitost modela. Naposljetku su 3 dubinske what-if analize pokušale unaprijediti nedostatne performanse. Koristili su se napredni koncepti iz teorija redova čekanja te se dao konačan zaključak o uspješnosti navedenih sustava kao i mogućnostima uporabe navedenog modela u praksi, i/ili njegovih what-if scenarija.

Ključne riječi: diskretna simulacija, teorija redova, mreža čekanja

Discrete simulation of queues

ABSTRACT

The paper initially goes through a basic theoretical background that will help in understanding the rest of the paper. After that, previous research based on discrete queue simulations is discussed. Next, more detailed theoretical explanations, flaws and disadvantages of discrete simulations are discussed. Also, the theory of queues is discussed, which is the practical part of this work. In a practical example using Python, a queue system was modeled on the example of a commercial bank. A simulation consisting of 500 iterations was performed and the effectiveness of the model was tested. Finally, 3 in-depth what-if analyzes tried to improve the lack of performance. Advanced concepts from queuing theories were used, and a final conclusion was given on the success of the mentioned systems, as well as the possibilities of using the mentioned model in practice, and/or its what-if scenarios.

Keywords: discrete simulation, queuing theory, queuing network

SADRŽAJ

1. Uvod

Za potrebe izrade diskretnih simulacija razvijeni su programski jezici i alati. Zajednička karakteristika je gledanje na model kao na niz slučajnih događaja gdje svaki uzrokuje promjenu stanja. Najveći broj razvijenih modela za diskretnu simulaciju napravljen je upravo zato da bi omogućio donošenje odluka, osobito odabir najpovoljnijih organizacijskih rješenja.

Usko povezano uz diskretne simulacije su i teorije redova. Teorije redova omogućuju analizu i optimizaciju redova čekanja, nečega što je iznimno važno kod dizajniranja novih projekata i poboljšanja postojećih. Kada se diskretne simulacije i teorije redova povežu, dobije se kvalitetan radni okvir kojim se mogu simulirati i prikazati stanja veoma slična „stvarnom svijetu“. Na temelju tih simulacija i analiza donose se zaključci.

Cilj rada je nakon teorijske obrade diskretnih simulacija i teorije redova čekanja, praktičnim prikazom i analizom istražiti redove čekanja klijenata u jednoj poslovnoj banci.

U ovom diplomskom radu prikazao se proces dolaska klijenata u generičku poslovnu banku.

Rad se sastoji od osam međusobno povezanih poglavlja. Nakon uvoda u kojem su predstavljeni ciljevi i svrha rada, slijedi poglavlje teorijska podloga i prethodna istraživanja gdje su definirani pojmovi poslovne i diskretne simulacije te ukratko objašnjen Python i biblioteka Ciw. Poglavlje metodologija rada daje uvid u metode koje se koriste u rješavanju identificiranih problema i postizanja ciljeva rada.

Poslije metodologije slijedi teorijski dio rada gdje se pobliže objašnjavaju diskretne simulacije i teorije redova čekanja. Nakon toga, dolazi se do praktičnog dijela rada.
U praktičnom dijelu postavljen je model jedne poslovne banke. Na temelju dobivene simulacije i what-if analiza dane su preporuke i komentari. Diplomski rad završava zaključkom.

2. Teorijska podloga i prethodna istraživanja

Jedno od područja istraživanja ovoga rada su diskretne simulacije, odnosno poslovne simulacije pomoću kojih bi poslovni subjekti trebali unaprijediti svoje poslovanje što ujedno predstavlja i svrhu ovog rada. Simulacija predstavlja imitaciju neke stvarne pojave, stanja ili procesa. Vagner (2006) navodi da pojam simulacije ima široko značenje i primjenu iako se taj pojam danas uglavnom veže samo uz zabavu i računalne igrice. „Poslovne simulacije opisuju različite ekonomske sustave te se koriste za prognoziranje ponašanja tih sustava. Simulacije te vrste počele su se intenzivnije koristiti prije 50-ak godina i danas su nezaobilazna metoda u razvijanju ljudskih potencijala. Također se koriste i za ispitivanje različitih strategija, analizu podataka i kao potpora odlučivanju" (Vagner, 2006:15). Važno je naglasiti da je jako važno obilježje poslovnih simulacija efikasno korištenje vremena. Uz pomoć simulacija možemo dobiti sve potrebne informacije u kratkom roku, za razliku od analize u stvarnom svijetu. Prema mišljenju Vagnera, „Područja primjene simulacija rastu svakim danom. Simulacije postaju sve realističnije, a matematički modeli na kojima se temelje postaju sve kompleksniji. Danas se na Internetu može naći više desetaka virtualnih svjetova, koji nastoje u svakom pogledu simulirati naš svakidašnji život. Neki idu tako daleko da tvrde kako je i ovaj svijet samo obična simulacija koju su pokrenuli ljudi iz budućnosti" (Vagner, 2006:17).

Diskretna simulacija (eng. *Discrete event simulation* (DES)) je proces kodiranja ponašanja složenih sustava kao slijeda čvrsto definiranih događaja gdje događaj predstavlja specifičnu promjenu u stanju sustava u nekom posebnom vremenskom trenutku. Događaj može nastupiti zbog ulaska ili izlaska entiteta u sustavu ili zbog promjene vrijednosti atributa. Događaji mogu biti uvjetni i bezuvjetni. Uvjetni su oni događaji koji se mogu dogoditi tek kada se ispuni neki određeni uvjet. Bezuvjetni, koji se naziva još i planirani, predstavlja one događaje koji se odvijaju nakon protoka određenog vremena. Božikov (2006) navodi da se simulacijom diskretnih događaja opisuju promjene stanja sustava koje se događaju samo u nekim vremenskim trenutcima. Modeli sadrže objekte određenih svojstava koji svojim međudjelovanjem u aktivnostima uzrokuju promjene stanja sustava u vremenu. Diskretna simulacija opisuje promjene stanja koje se odvijaju diskontinuirano u vremenu. Namijenjena je razvoju modela koji detaljno opisuju strukturu sustava i njegove elemente, tj. oponašaju stvarne sustave i procese, te objekte iz stvarnog svijeta i njihovo međudjelovanje. Glavni elementi diskretne simulacije su entiteti, atributi i događaji, te vrijeme pomaka. Entiteti (objekti) ovih modela mogu biti stalni ili privremeni i imaju atribute. Stalni entiteti (ili resursi) ostaju u modelu tijekom čitavog vremena trajanja simulacije dok su privremeni entiteti oni koji prolaze kroz sustav. Atributima entiteta opisuju se njihova svojstva (svaki entitet može imati više atributa). Definiraju se klase entiteta i skupovi entiteta (grupe entiteta pojedine klase koji imaju neka zajednička svojstva). S obzirom na mogućnosti dinamičke promjene svojstava, entiteti se mogu tijekom simulacije premještati iz skupa u skup, ali i mijenjati vrijednosti atributa.

Diskretna simulacija se najčešće koristi za modeliranje i analizu sustava s repovima čekanja na resurse. Repovi čekanja čine grupu privremenih entiteta koji čekaju da se oslobodi neki resurs. Koristeći se ovim pojmovima, može se diskretna simulacija ukratko opisati ovako: entiteti koji imaju atribute međudjeluju u aktivnostima uz određene uvjete stvarajući događaje koji mijenjaju stanje sustava. Treba naglasiti da se kod diskretne simulacije i vrijeme mijenja diskretno, točnije diskontinuirano. Od trenutka u kojem se dogodio posljednji događaj, vrijeme ide na trenutak u kojem će se dogoditi sljedeći događaj.

Kod diskretne simulacije, nezavisne (ulazne varijable) mogu biti:

- vrijeme između dva uzastopna dolaska entiteta
- vrijeme odvijanja pojedinog procesa (npr. vrijeme posluživanja na šalteru)
- broj raspoloživih resursa (npr. broj šaltera, ili djelatnika, ili strojeva)
- organizacija čekanja (FIFO, LIFO, po prioritetu)

Dok zavisne (izlazne) varijable čije se vrijednosti dobiju kao rezultat simulacije su

- dužina reda čekanja (npr. broj pacijenata u čekaonici)
- vrijeme čekanja u redu (vrijeme koje entitet provede u čekanju)
- iskorištenje resursa – postotak iskorištenja resursa (npr. % radnog vremena djelatnika u kojem on radi)
- propusnost (broj entiteta koje sustav može obraditi)

Programski jezik *Python* pruža maksimalnu fleksibilnost i široku pokrivenost situacija s njegovim brojnim bibliotekama koje omogućavaju modeliranje prema željenoj razini detalja i složenosti. *Ciw* simulacijska biblioteka, koja je napisana u *Pythonu*, omogućuje diskretnu simulaciju sustava s otvorenim redovima čekanja. Za izvođenje simulacije najčešće se koristi pomak vremena na sljedeći događaj. Znači da nakon izvođenja jednog događaja pronalazi se događaj koji je sljedeći na redu. Vrijeme simulacije se pomiče na vrijeme izvođenja tog novog događaja. Novi se događaj izvede, tom prilikom se događa i promjena stanja sustava, točnije mijenjaju se atributi jednog ili više entiteta. Slijedi ponavljanje koraka 1-3 sve dok više nema događaja za izvesti ili je pak ispunjen neki unaprijed utvrđeni uvjet da simulacija završava.

Vrednovanje simulacijskog modela ima za cilj eliminaciju različitih vrsta grešaka modela, koje mogu biti (Čerić, 2013):

- greške u logici modela
- greške u matematičkim relacijama
- greške u programu
- greške u ulaznim podacima
- greške u načinu korištenja modela te
- greške u obradi i interpretaciji rezultata simulacijskih eksperimenata

Brojna su istraživanja provedena na temu diskretnih simulacija redova čekanja. Neka od prijašnjih istraživanja koja se tiču diskretnih simulacija unutar hrvatske znanstvene zajednice tiču se, primjerice, simulacije prodaje pretplatničkih karata javnoga prijevoza (Kaurić i Pogarčić, 2013). U tom radu je opisan promatrani sustav prodaje te je napravljen model i isti je vizualiziran. Rad je prikupio podatke o vremenu dolaska i odlaska korisnika, vremenu čekanja u redu te koliko je usluga trajala. Na temelju tih podataka, dan je i zaključak kako postojeća dva šaltera nisu dovoljna te da bi se otvaranjem dodatnog šaltera smanjio red čekanja u vremenu najvećih gužvi.

Simulacija diskretnih događaja primjenjuje se u mnogim domenama kako bi se riješili problemi. Kako navode Collins et al. (2021), s primjenom u obrani, preko prijevoza, do zdravstvene skrbi, diskretna simulacija pomaže korisnicima da razumiju složena pitanja i nauče kako se nositi s povezanim problemima. Promjenom u zahtjevima klijenata te tehnološkim napretkom došlo je i do promjena u diskretnim simulacijama.

3. Metodologija rada

Različite metodologije su korištene prilikom pisanja ovog diplomskog rada. Od izrazitog značaja jest bila metoda indukcije, budući da se do konačnog zaključka došlo praćenjem individualnih grafova te se tim stvarala opća slika o stanju simulacije. Suprotno od toga, dedukcija je pomogla da se dođe do konkretnih pojedinosti za određenu simulaciju. Metoda deskripcije je najzastupljenija u teorijskom dijelu diplomskog rada, kada se govorilo o diskretnim simulacijama, teoriji redova i mreži redova. Metodom deskripcije došlo se do primjera koji približavaju situacije u kojima su simulacije i teorije redova prisutne. Povijesna metoda koristila se za prikupljanje različitih istraženih činjenica kojima se dolazilo do konkretnih formula i opisa simulacija i redova čekanja.

Vrlo važna metoda jest bila metoda analize, kojom su se uočavale posljedične veze između ponašanja entiteta u sustavu. Analiziranjem grafova dolazilo se do bitnih zaključaka kojim se mogao postaviti pravilan model za what-if scenarije.

U izradi ovoga rada korišteno je elektroničko računalo, Python, Office paket s Microsoft Wordom i PowerPointom, stručna literatura preuzeta s Interneta u obliku .pdf datoteka i literatura posuđena iz gradske knjižnice. Grafovi su nacrtani pomoću Python biblioteke matplotlib.

4. Diskretne simulacije

Kako bi se shvatilo što su to diskretne simulacije, bitno je objasniti i općenito simulaciju kao pojam. Prema Stewartu (2014), simulacija je imitacija sustava, pri čemu imitacija predstavlja kopiranje nečega ili nekoga. Postoje dvije vrste simulacija: *dinamičke* i *statičke*. Statičke simulacije imitiraju sustav koji je fiksirana točka u vremenu, a dinamičke simulacije oponašaju sustav koji teče kroz vrijeme. Ono čemu je ovaj rad posvećen jest računalnoj dinamičkoj simulaciji sustava koji protječe kroz vrijeme pomoću diskretnih simulacija događaja.

Diskretna simulacija će se promatrati prema Banks i dr. (2010), koji navode da je to simulacija modela sustava u kojima se varijabla stanja mijenja samo u diskretnom skupu točaka u vremenu. Simulacijski modeli analizirani su numeričkim metodama, a ne analitičkim metodama. Analitičke metode koriste deduktivno razmišljanje matematike kao "rješenje" modela. Na primjer, diferencijalni račun može se koristiti za izračunavanje politike minimalnih troškova za neke modele inventara. Numeričke metode koriste računalne postupke za "rješavanje" matematičkih modela. U slučaju simulacijskih modela, koji koriste numeričke metode, modeli se "pokreću" umjesto da se rješavaju - to jest, umjetna povijest sustava generira se iz pretpostavki modela, a opažanja se prikupljaju da bi se analizirao i procijenio pravi sustav mjerila performansi. Simulacijski modeli u stvarnom svijetu prilično su veliki, a količina podataka koji se pohranjuju i kojima se manipulira je ogromna, pa se takva ispitivanja obično provode uz pomoć računala. Međutim, mnogo se uvida može dobiti ručnom simulacijom manjih modela (Banks i dr., 2010: 20-23).

4.1. Prednosti i nedostaci simulacija

Simulacija je privlačna klijentu jer oponaša ono što se događa u stvarnom sustavu ili ono što se percipira za sustav koji je u fazi projektiranja. Izlazni podaci iz simulacije trebaju izravno odgovarati izlazima koji se mogu uočiti i mjeriti u stvarnom sustavu. Dodatno, moguće je razviti simulacijski model sustava bez dvojbenih pretpostavki (kao što je ista statistička distribucija za svaku slučajnu varijablu) matematički rješivih modela. Zbog ovih i drugih razloga, simulacija je često tehnika izbora u rješavanju problema.

Za razliku od optimizacijskih modela, simulacijski modeli se više "pokreću" nego rješavaju. S obzirom na određeni skup ulaznih podataka i karakteristika modela, model se pokreće i promatra se simulirano ponašanje. Ovaj proces mijenjanja inputa i karakteristika modela rezultira skupom scenarija koji se procjenjuju. Dobro rješenje, bilo u analizi postojećeg sustava ili u dizajnu novog sustava, tada se preporučuje za implementaciju.

Simulacija ima mnoge prednosti, ali i neke nedostatke. Prema Banks i dr. (2010), neke prednosti su sljedeće:

1. Nove politike, operativni postupci, pravila odlučivanja, tokovi informacija, organizacijski postupci, itd. mogu se istražiti bez ometanja tekućih operacija stvarnog sustava.
2. Novi hardverski dizajn, fizički rasporedi, transportni sustavi itd. mogu se testirati bez izdvajanja resursa za njihovu nabavu.
3. Hipoteze o tome kako ili zašto se određeni fenomeni pojavljuju mogu se testirati na izvedivost.
4. Vrijeme se može komprimirati ili proširiti kako bi se omogućilo ubrzanje ili usporavanje fenomena pod istragom.
5. Može se dobiti uvid u interakciju varijabli.
6. Može se dobiti uvid o važnosti varijabli za performanse sustava.
7. Može se provesti analiza uskih grla kako bi se otkrilo gdje se radovi u tijeku, informacije i materijali prekomjerno odgađaju.
8. Analiza simulacije može pomoći u razumijevanju kako sustav radi, a ne kako pojedinci misle da taj sustav funkcionira.
9. Na „*what-if*“ pitanja se može odgovoriti. Ovo je osobito korisno u dizajnu novih sustava.

Neki od nedostataka su sljedeći:

1. Izrada modela zahtijeva posebnu obuku. To je umjetnost koja se uči s vremenom i iskustvom. Nadalje, ako dva modela konstruiraju različiti kompetentni pojedinci, oni bi mogli imati sličnosti, ali je vrlo malo vjerojatno da će biti isti.
2. Rezultate simulacije može biti teško protumačiti; većina simulacijskih izlaza su u biti slučajne varijable (obično se temelje na slučajnim ulazima), tako da može biti teško razlikovati je li opažanje rezultat međuodnosa sustava ili slučajnosti.
3. Simulacijsko modeliranje i analiza mogu biti dugotrajni i skupi. Štednja na resursima za modeliranje i analizu mogla bi rezultirati simulacijskim modelom ili analizom koji nisu dostatni za zadatak.

4. Simulacija se koristi u nekim slučajevima kada je analitičko rješenje moguće, ili čak poželjno.

Prema Banks i dr. (2010:24), ova četiri nedostatka mogu se kompenzirati na sljedeći način:

1. Proizvođači simulacijskog softvera aktivno razvijaju pakete koji sadrže modele kojima su za rad potrebni samo ulazni podaci. Takvi modeli imaju generičku oznaku "simulator" ili "predložak".
2. Mnogi proizvođači simulacijskih softvera razvili su mogućnosti analize izlaza unutar svojih paketa za izvođenje vrlo temeljite analize.
3. Simulacija se danas može izvesti brže nego jučer, a bit će još brža sutra, zbog napretka u hardveru koji dopušta brzo izvođenje scenarija i zbog napretka u mnogim simulacijskim paketima. Na primjer, neki softver za simulaciju sadrži konstrukcije za modeliranje rukovanja materijalom koji koristi takve transportere kao što su viličari, pokretne trake i automatizirano vođena vozila.
4. Modeli zatvorenog oblika nisu praktični za analizu većine složenih sustava kojim se analitičari susreću u praksi. U višegodišnjoj savjetodavnoj praksi dvojice autora nije se susreo niti jedan problem koji bi se mogao riješiti zatvorenim rješenjem.

Banks i dr. (2010) navode neke primjere sustava koji se mogu simulirati:

- Proizvodni sustavi
- Javni sustavi: zdravstvo, vojska, prirodni resursi
- Transportni sustavi
- Konstrukcijski sustavi
- Sustavi restorana i zabave
- Reinženjering/upravljanje poslovnim procesima
- Prerada hrane
- Performanse računalnog sustava

Iako nije u popisu primjera, uslužni sustav se isto tako može simulirati, specifičnije: uslužni sustav poslovne banke, što je i tema praktičnog dijela ovog rada.

4.2. Izgradnja simulacijskih modela

U simulacijskom projektu prvi korak je analiza stvarnosti u kojoj se problem nalazi. Ova stvarnost je predstavljena kao sustav. Sustav je skup elemenata i odnosa među njima. Najrelevantniji aspekt sustava je da je on nešto više od cjeline njegovih dijelova. Odnosno, njegovo se ponašanje ne može u potpunosti spoznati samo pojedinačnom analizom ponašanja entiteta. Struktura i priroda odnosa između dijelova određuje, na ponekad 'naizgled' nepredvidive načine, ukupno ponašanje sustava.

Kao što se istaknulo, sustav nije stvarnost nego njezin prikaz, u konačnici on je mentalna konstrukcija. Potrebno je odabrati razinu detalja sustava koje je potrebno definirati. Ova će razina ovisiti o ciljevima studija, odnosno o vrsti problema kojeg se želi riješiti. Na primjer, ako se namjerava proučavati promet privatnih vozila kako bi se definirale politike za smanjenje emisija i vremena prijevoza, boja kao ni godina proizvodnje automobila nisu relevantni. Potonji bi podatak mogao biti zanimljiv ako bi se analiziralo zagađenje koje proizvodi promet; u ovom slučaju, zapravo, godina proizvodnje može pružiti informacije o emisijama motora (Stewart, 2014:77-82).

Općenito, dobro pravilo je da je razina detalja minimalna, kompatibilna s odgovorima koji se žele dobiti iz simulacije. Obično se kreće od jednostavnog modela, s nekoliko elemenata, a zatim ga se usavršava, postupno povećavajući stupanj detalja, dok se ne dođe do modela koji je zadovoljavajući s obzirom na ciljeve.

Vrlo važan element je vrsta formalne notacije koja se koristi za predstavljanje i opisivanje sustava. Prije svega, oznake moraju biti dovoljno jednostavne i jasne da ih mogu lako razumjeti svi oni koji su eksplicitno ili implicitno uključeni u proces modeliranja, odnosno oni koji se često nazivaju dionicima. Pitanje je ne samo tko će imati zadatak donositi konačne odluke i provoditi ih, nego i tko će ih onda morati konkretno provesti u djelo, te tko, djelujući u promatranoj stvarnosti, posjeduje takvo bitno znanje za definiranje i modeliranje sustava kao i podatke potrebne za eksperimentiranje. Lako razumljiv model omogućuje svakome provjeru njegove ispravnosti, a time i veću učinkovitost i djelotvornost procesa modeliranja. Osim što jamče jasnoću i jednostavnost, notacije također moraju biti dovoljno rigorozne da omoguće brzu i učinkovitu implementaciju modela pomoću odabranog softvera.

U ovom odlomku opisat će se formalni elementi koji će se koristiti za karakterizaciju sustava, također uvodeći oznake koje će se najčešće koristiti. Pozornost će biti usmjerena na fizički sustav koji se želi modelirati, na elemente koji se u njemu pojavljuju, na odnose između tih elemenata i aktivnosti koje se tamo odvijaju. Zatim će se pregledati proces modeliranja s ciljem implementacije simulacije predmetnog sustava.

Glavni elementi koji će se koristiti za predstavljanje sustava su entiteti i operacije. Prvi opisuju ono što karakterizira sustav sa statične točke gledišta, dok drugi omogućuju opis njegove evolucije tijekom vremena i istaknutost odnosa između njegovih dijelova (Stewart, 2014:82-87).

4.2.1. Entiteti

Osnovni objekti sustava koji se modeliraju su entiteti. To su elementi sustava koji se razmatraju pojedinačno, a o čijem statusu se čuva informacija tijekom simulacije. Tipični entiteti su pacijent koji se pojavi na recepciji bolnice, komad materijala koji se obrađuje na tekućoj traci ili avion koji čeka slijetanje u zračnoj luci.

U nastavku valja razlikovati tipove entiteta, odnosno klase, i pojedinačne entitete ili objekte. Kroz klasu se definira apstraktni tip entiteta sa svojim svojstvima. Svojstva se sastoje od skupa atributa koji karakteriziraju sve entitete koji su instance te klase.

Na primjer, u sustavu koji se sastoji od poštanskog ureda kojem se korisnici obraćaju za određene vrste usluga, može se definirati klasa korisnika koju karakteriziraju atributi kao što su: vrsta i količina tražene usluge, vrijeme dolaska na šalter.
S druge strane, objekt klase bit će pojedini kupac, odnosno, primjerice, kupac koji se pojavi na šalteru u 11:32 s dvije potvrde o tekućem računu za plaćanje. Stoga će tip entiteta odgovarati jednoj klasi i varijabilnom broju objekata, ograničenom ili neograničenom. Naravno, stvarnost može biti složenija, na primjer s više podklasa iste klase.

Objekti se ponekad mogu grupirati u skupove. Na primjer, u bolničkoj hitnoj pomoći mogu se uzeti u obzir sve medicinske sestre. Tijekom rada sustava, svaka aktivnost može zahtijevati jednu ili više medicinskih sestara; ako broj slobodnih medicinskih sestara nije dovoljan za obavljanje djelatnosti, tada ona neće moći započeti dok se ne oslobodi onoliko medicinskih sestara koliko je potrebno.

Neki entiteti su kolektivni. Na primjer, parkiralište sa 100 parkirnih mjesta može se predstaviti klasom "parkirno mjesto" sa 100 primjeraka, odnosno objekata tipa "parkirno mjesto". No, osim u vrlo posebnim situacijama, neće biti zanimljivo za promatrati je li jedno parkirno mjesto slobodno ili zauzeto; samo se treba znati ima li slobodnih mjesta na parkingu i koliko ih ima.
U ovom slučaju parkiralište se smatra jednim kolektivnim entitetom: tada se kaže da je to resurs. Očito, odluka hoće li se određeni entitet smatrati resursom izbor je koji ovisi o projektantu simulacije: čak se i pojedinačni entitet može smatrati resursom s kardinalnošću 1. Općenito, to je izbor koji se mora voditi se razmatranjima vezanim uz globalnu ekonomiju modela koji se gradi.

Entiteti mogu biti trajni ili privremeni, aktivni ili pasivni. Na primjer, u slučaju reda na kazališnoj blagajni, korisnici koji dođu u red i napuste sustav nakon što su usluženi, mogu se smatrati privremenim i pasivnim entitetima, dok blagajnici imaju ulogu stalnih i aktivnih. Međutim, razlika je ponekad proizvoljna, ovisno o percepciji koju graditelj modela ima o sustavu i njegovim izborima.
Konačno, entitet može biti u stanju čekanja ili može biti zauzet obavljanjem neke aktivnosti. Stanje sustava se definira kao skup stanja entiteta koji ga čine (Pidd, 2004:63-65).

4.2.2. Operacije

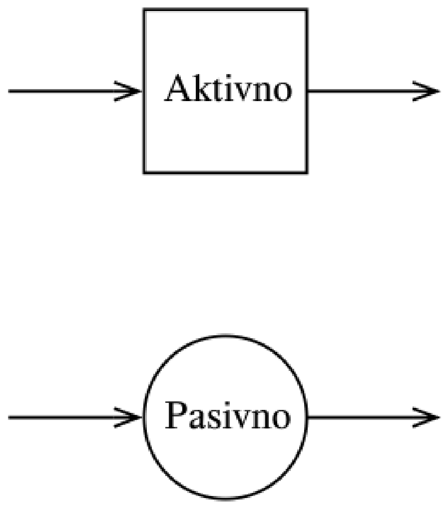
Pod pojmom operacija označava se sve ono što se odnosi na dinamiku sustava i što uzrokuje njegovu evoluciju prelazeći iz jednog stanja u drugo. Posebno će se govoriti o događajima, aktivnostima i procesima.
Događaji. U simulacijskim tekstovima termin događaj vrlo često označava trenutak vremena u kojem se događa promjena stanja sustava. Korisnije ga je ovdje shvatiti kao činjenicu koja proizvodi promjenu stanja u sustavu. Na primjer, završetak usluge korisnika na šalteru uzrokuje izlazak kupca iz sustava, a zaposlenik na šalteru prelazi iz zauzetog u slobodan.
Aktivnosti. Svaki od entiteta/objekata prisutnih u sustavu u bilo kojem trenutku provodi aktivnosti, uzimajući u obzir i čekanje kao posebnu aktivnost. Aktivnost je nešto što se odvija između dva događaja i odgovara stanju jednog ili više entiteta. Na primjer, uslužna aktivnost na šalteru odvija se između događaja 'početka usluge' i događaja 'kraja usluge', a uključuje i

zaposlenika koji radi na šalteru i kupca koji je uslužen. Prvi ima aktivnu ulogu u ovoj aktivnosti, dok drugi ima pasivnu ulogu.

Procesi. Procesi su unaprijed definirani nizovi ili ciklusi aktivnosti (a time i događaja). Na primjer, putnik u zrakoplovu prolazi kroz sljedeći slijed aktivnosti: dolazak u check-in red, čekanje u redu, check-in, dolazak na sigurnosnu provjeru, čekanje u redu, sigurnosna provjera, čekanje poziva za ukrcaj, provjera na ulazu za ukrcaj, ukrcaj na avion.

4.2.3. Ciklusi aktivnosti

Široko korištena metoda za opisivanje prijelaza iz jednog stanja u drugo u sustavu sastoji se od takozvanih ciklusa aktivnosti. U ciklusu aktivnosti, stanja su predstavljena kao čvorovi u grafu, a prijelazi iz jednog stanja u drugo kao usmjereni bridovi, odnosno lukovi. Razlikuju se aktivna i pasivna stanja pomoću dvije različite vrste grafičkog prikaza čvorova: kvadrati za predstavljanje aktivnih stanja i kružići za predstavljanje pasivnih stanja (slika 1). Često se dvije vrste stanja izmjenjuju u istom ciklusu: aktivno stanje slijedi pasivno i obrnuto.

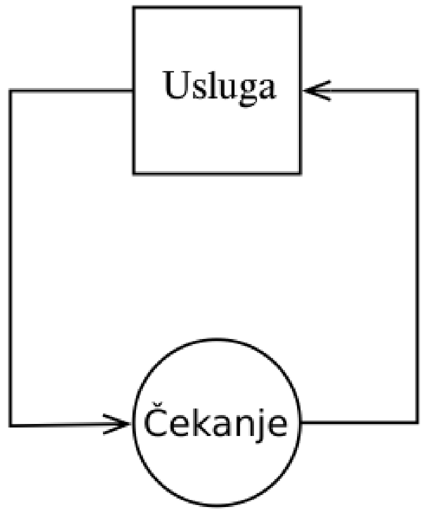


Slika 1. Prikaz aktivnog i pasivnog stanja čvorova. Izvor: Gallo, G. (2007)

Vraćanjem na primjer već korištene šalterske usluge, postoje dvije klase, klasa *blagajnik* i klasa *kupac*.

Pretpostavit će se da prvi ima samo jednu instancu, pretpostavivši postojanje samo jedne grane, dok drugi ima *a priori* neograničen broj instanci. Objekti klase kupca su privremeni entiteti u ovom sustavu; zapravo, oni ulaze u sustav kada dođu u ured i nestaju iz sustava nakon što su usluženi. Suprotni entitet je umjesto toga stalni entitet (Pidd, 2004:66-67).

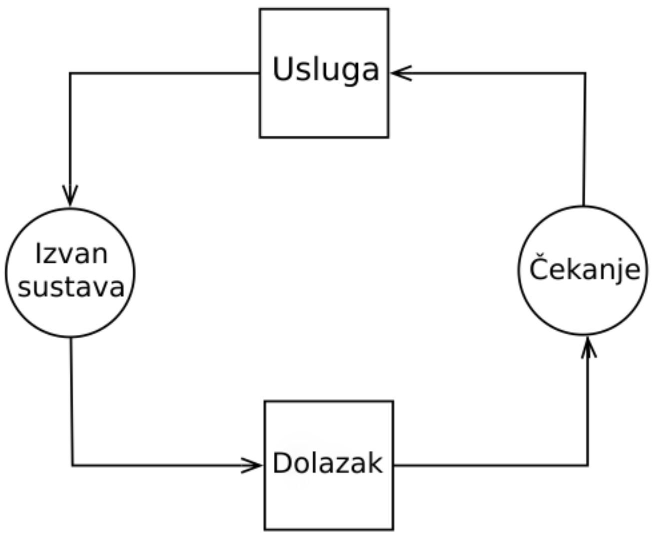
Ciklus stanja blagajnika na šalteru je prikazan u nastavku.



Slika 2. Prikaz ciklusa stanja blagajnika na šalteru. Izvor: Gallo, G. (2007)

Može se vidjeti kako blagajnik ima dva stanja: Usluga i Čekanje.

Ciklus aktivnosti omogućuje da se jasno istaknu ne samo stvarna stanja, već i aktivnosti povezane s njima kao i događaji koji označavaju prijelaz iz jednog stanja u drugo.
U nastavku slijedi prikaz ciklusa aktivnosti kupca.



Slika 3. Prikaz ciklusa aktivnosti kupca. Izvor: Gallo, G. (2007)

U ovom primjeru postoji samo jedna vrsta kupaca i to čini poslovni ciklus posebno jednostavnim. Nešto složeniji primjer je onaj „kazališne blagajne" koju koristi Pidd (2004). U ovom primjeru sustav se sastoji od blagajne kazališta gdje se prodaju ulaznice. Službenik na blagajni (posluživanje), osim što prodaje ulaznice kupcima, dužan je i odgovarati na telefonske pozive davanjem traženih podataka. Postoje, dakle, dva reda, fizički red kupaca, ispred šaltera, i virtualni, sastavljen od poziva na čekanju (telefonski sustav bi trebao biti dovoljno sofisticiran da to omogući). Oba reda čekanja obrađuju se FIFO politikom i klijenti uvijek imaju prioritet nad pozivima. U ovom slučaju imamo tri klase entiteta: korisnik (neograničen broj privremenih i pasivnih entiteta), poziv (neograničen broj privremenih i pasivnih entiteta), blagajna (jedan entitet, stalni i aktivni) (Pidd, 2004:68-78).

Iako se već u ovom poglavlju spominju redovi čekanja, sljedeće poglavlje posvećeno je detaljnijoj obradi teorije redova kako bi se naposljetku mogao postaviti kvalitetan model i adekvatna simulacija u Pythonu.

5. Teorija redova

Teorija redova (ili redova čekanja) predstavlja analizu fenomena čekanja koji se mogu pojaviti u slučaju potražnje za uslugom. Zapravo, u mnogim svakodnevnim situacijama sama potražnja za uslugom ili mogućnost pružanja iste podložni su neizvjesnosti: to je slučaj zahtjeva koji se javljaju nasumično i neovisno jedan o drugome. Oni koji nude usluge su u nemogućnosti odmah udovoljiti traženim zahtjevima.

Postoje brojni slučajevi iz svakodnevice:

- Klijenti u banci
- Ljudi koji čekaju taksi
- Automobili na raskrižju
- Zrakoplovi koji čekaju polijetanje ili slijetanje
- Dijelovi koji čekaju na obradu
- Ljudi u hitnoj službi.

Sve su to situacije u kojima je primjenjiva teorija čekanja.

Studije i simulacije provode se u mnogim slučajevima: za sinkronizaciju semafora na temelju prometa u tom području ili za ispis procjene vremena čekanja na kartici za neki ured ili bolnicu. Međutim, ne mogu se sve te situacije opisati istim slučajnim varijablama i istim zakonima vjerojatnosti. Stoga se posebna pozornost mora usmjeriti na vrstu uvedenih slučajnih varijabli (Sztrik, 2021: 9-10).

5.1. Proces čekanja

S praktičnog stajališta, red čekanja je sustav sastavljen od nepraznog skupa poslužitelja koji nude uslugu korisnicima (kupcima) koji pripadaju populaciji sastavljenoj od vjerojatnih korisnika i koji, ako čekaju na uslugu, organizirano sami stoje u redu čekanja u vrsti međuspremnika (engl. *bufferu*).

Odabir korisnika iz reda čekanja koji će prvi koristiti uslugu odvija se u skladu sa specifičnom *uslužnom disciplinom.*

Dolazak kupaca je nasumičan i upravo zbog takve neizvjesnosti čak i kada priljev istih nije veći od raspoloživog kapaciteta sustava, nastaju redovi.

Parametri koji opisuju veću ili manju sposobnost sustava da zadovolji zahtjeve korisnika i koji određuju kretanje reda su:

• Broj poslužitelja i

• Vrijeme usluge.

Općenito, međutim, postoji nekoliko parametara koji se koriste za potpuni opis reda čekanja, a to su:

•t_a: vremenski interval između dva uzastopna cilja;

• t_s: vrijeme usluge za i-tog kupca;

• ρ: faktor iskorištenja koji predstavlja omjer između vremena provedenog u službi i dostupnog vremena u cjelini

• W: ukupno vrijeme koje je generički kupac proveo u redu prije nego što je uslužen;

• W_s: ukupno vrijeme provedeno od strane generičkog korisnika u sustavu;

• n: broj korisnika u sustavu u trenutku razmatranja (status sustava);

• l_s: prosječan broj korisnika u sustavu;

• l_q: prosječan broj kupaca koji čekaju na uslugu;

• ponašanje korisnika nakon usluge;

• l: maksimalna duljina reda čekanja, ako postoji.

• disciplina usluge: pravilo prema kojem se odabiru kupci koji dolaze na red usluživanja;

Discipline usluge uvelike variraju u svakodnevnim slučajevima, neki primjeri su navedeni u nastavku:

1. FIFO / FCFS, odnosno „first-in, first-out“ i „first-come, first-served“. Odnosno, klijenti se poslužuju prema redoslijedu dolaska.
2. LIFO / LCFS, „last-in, first-out“, „last-come, first-served“. Trivijalan primjer je oprano i složeno posuđe, posljednje dodano prvo će se uzeti. Ova se disciplina proučava u skladištima kako se zadnji pripremljeni proizvodi ne bi ujedno i prvi prodali, inače će prvi dodani ostati tamo dok se skladište ne isprazni.
3. SIRO, „service in random order“. Ovo je usluga koja se temelji na redoslijedu prioriteta, tipična za prvu pomoć (Sztrik, 2021:10-12).

5.1.1. Stohastički procesi

Stohastički procesi su matematički modeli prikladni za proučavanje trendova pojava koje slijede slučajne ili vjerojatnosne zakone. Oni su vrlo korisni za analizu prirodnih pojava u kojima se slučajna komponenta uvijek može smatrati prisutnom zbog same njihove prirode, ali i zbog pogrešaka u promatranju. To znači da rezultat promatranja u bilo kojem trenutku nije siguran podatak i bez pogrešaka povezanih s njim, ali će uvijek imati povezanu nesigurnost. Iz tog razloga, svaka od njegovih vrijednosti bit će najvjerojatnija među onima koje se mogu pretpostaviti.

Dva su stohastička procesa koja karakteriziraju sustav čekanja:

1. Proces dolazaka: ovo je karakterizirano distribucijom vjerojatnosti vremena između dolazaka, tj. vremena između dolazaka dvaju uzastopnih kupaca. Kako bi se olakšalo proučavanje, u teoriji čekanja u redu, oni se smatraju stacionarnim, tj. njihova statistička svojstva (kao što je prosječno vrijeme između dolaska) ne variraju tijekom vremena.
2. Proces usluge: distribucija vjerojatnosti opisuje vrijeme potrebno svakom operateru da zadovolji zahtjeve generičkog korisnika. Raspodjela vremena dolaska i usluge može biti bilo koje vrste, na primjer, kupaca koji ulaze u trgovinu. Ulazi mogu biti načelno nasumični ako kupci nisu vezani osobnim odnosima, već ih često čini više ljudi zajedno kao obiteljska jedinica ili prisutnost većeg broja ljudi u trgovini može pobuditi znatiželju kod drugih prolaznika koji će ući da vide koji su to artikli zanimljivi.

(Sztrik, 2021:10-12).

5.1.2. Proces rađanja i umiranja

Eksponencijalna usluga te Poissonovi dolasci čine modele čekanja Markovljevim procesima. Temeljni Markovljev proces koji predstavlja broj kupaca u takvim sustavima poznat je kao proces rađanja i umiranja, često korišten u populacijskim modelima. Pojmovi rođenje-smrt odnose se na povećanje, odnosno smanjenje populacije. Kada se govori o sustavima čekanja, to bi bili dolasci i odlasci.

Zadržavajući terminologiju rođenje (dolazak) – smrt (odlazak), kada je veličina populacije n, neka λ_n i μ_n budu infinitezimalne stope prijelaza (generatori) rođenja i smrti. Kada je populacija broj kupaca, λ_ni μ_n pokazuju da stope dolaska i usluge ovise o broju kupaca u sustavu (Narayan Bhat, 2015:37).

Generalizirajući svojstva Poissonovog procesa, mogu se dati sljedeće izjave o vjerojatnosti prijelaza tijekom intervala (t, t + Δt].

Rođenje ($n \geq 0$):

$$\text{P(jedno rođenje)} = \lambda_n \Delta t + o(\Delta t)$$

$$\text{P(bez rođenja)} = 1 - \lambda_n \Delta t + o(\Delta t)$$

$$\text{P(više od jednog rođenja)} = o(\Delta t)$$

Smrt ($n > 0$):

$$\text{P(jedna smrt)} = \mu_n \Delta t + o(\Delta t)$$

$$\text{P(bez smrti)} = 1 - \mu_n \Delta t + o(\Delta t)$$

$$\text{P(više od jedne smrti)} = o(\Delta t)$$

Gdje je $o(\Delta t)$ takav da $\frac{o(\Delta t)}{(\Delta t)} \to 0$ kada $\Delta t \to 0$

Ukupna vjerojatnost triju događaja jednaka je 1.

Neka je $Q(t)$ broj kupaca u sustavu u trenutku t.

Definira se

$P_{in}(t) = P[Q(t) = n|Q(0) = i]$.

Uključujući vjerojatnosti prijelaza tijekom intervala (t, t+Δt], dobiva se

$P_{n,n+1}(\Delta t) = \lambda_n \Delta t + o(\Delta t)$, za n = 0, 1, 2, …

$P_{n,n-1}(\Delta t) = \mu_n \Delta t + o(\Delta t)$, za n =1, 2, 3, …

$P_{nn}(\Delta t) = 1 - \lambda_n \Delta t - \mu_n \Delta t + o(\Delta t)$, za n = 1, 2, 3, …

$P_{nj}(\Delta t) = o(\Delta t)$, pri čemu j ≠ n – 1, n, n + 1

Kako navodi Narayan Bhat (2015:38), infinitezimalne stope prijelaza dovode do sljedeće generatorske matrice za proces rađanja i umiranja u sustavu čekanja:

$$A = \begin{bmatrix} -\lambda_0 & \lambda_0 & & \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & \\ & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 \\ & & & \\ & & & \end{bmatrix}$$

Generatorska matrica A rezultira sljedećim Kolmogorovljevim jednadžbama za $P_{in}(t)$ (Prema Narayan Bhat (2015:38), od sada pa nadalje u ovoj sekciji će se pisati $P_{in}(t) \equiv P_n(t)$ te će se početno stanje *i* pisati po potrebi.):

$$P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t)$$

$$P_n'(t) = -(\lambda_n + \mu_n)P_n(t) + \lambda_{n-1}P_{n-1}(t) + \mu_{n+1}P_{n+1}(t)$$, pri čemu je n = 1, 2, …

Neka $\Delta t \to 0$.

Kako tvrdi Narayan Bhat (2015:39), eksplicitno deriviranje $P_n(t)$ je mukotrpan proces te je najšire korišten rezultat kada $t \to \infty$.

Kratko objašnjenje varijabli: P_n predstavlja graničnu vjerojatnost da će u sustavu biti točno n klijenata, a $P_n(t)$ vjerojatnost da se u sustavu u trenutku t nalazi n klijenata.

U stanju ravnoteže, ponašanje procesa je neovisno o vremenskom parametru i početnom stanju:

$\lim_{t \to \infty} P_{in}(t) = p_n$ za n = 0, 1, 2 …

te stoga je

$P_n'(t) \to 0$ kada $t \to \infty$.

Koristeći ove rezultate dobiju se sljedeće jednadžbe:

$$0 = -\lambda_0 p_0 + \mu_1 p_1$$

$$0 = -(\lambda_n + \mu_n)p_n + \lambda_{n-1}p_{n-1} + \mu_{n+1}p_{n+1}$$ za n = 1, 2, …

Te jednadžbe riješe se pomoću rekurzije te se, prema Narayan Bhat (2015:40) dobije sljedeće

$$p_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} p_0$$

Teorem 4.1.1.[1] kojeg navodi Narayan Bhat (2015) daje normalizacijski uvjet $\sum_{n \in S} p_n = 1$ koji kada se primijeni u jednadžbi za p_n dobije se

$$p_0 = \left[1 + \sum_{n=1}^{\infty} \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n}\right]^{-1}$$

Poznavanje izraza ovih vjerojatnosti i tumačenje procesa repa kao procesa rađanja i umiranja omogućuje dobivanje informacija o srednjem stanju repa, osobito u M/M/1 redovima koji imaju jedan red i samo jednog operatora.

[1] Pogledati (Narayan Bhat, 2015), str. 39, Theorem 4.1.1

5.1.3. Kendallova notacija

Kendallova notacija koristi se za karakterizaciju sustava čekanja. Dana je s A/B/c/x/y/z, gdje je:

A - distribucija vremena dolazaka jedinica u sustav

B - distribucija vremena usluživanja jedinica

c - broj poslužitelja

x - kapacitet sustava opsluživanja

y - veličina populacije iz koje klijenti dolaze u sustav

z - disciplina reda čekanja.

Neke moguće distribucije (za A i/ili B) su M (za Markovljevu), E (za Erlangovu), Ck (za Coxianovu reda k), G (za opću) i D za determinističku (ili konstantnu). Za opći proces dolaska, oznaka GI se ponekad koristi umjesto G kako bi se naznačilo da, iako raspodjela vremena među dolascima može biti potpuno opća, uzastopni dolasci su neovisni jedni o drugima. Broj poslužitelja (c) često se uzima kao 1. Ova prva tri parametra su uvijek navedena. Tako, na primjer, M/M/1 red čekanja znači da su i proces dolaska i proces usluge Markovljevi (iako obično kažemo da je proces dolaska Poissonov i distribucija vremena usluge je eksponencijalna) i postoji jedan poslužitelj. Slova koja određuju kapacitet sustava, populaciju korisnika i disciplinu raspoređivanja mogu se izostaviti uz razumijevanje da su zadane vrijednosti za kapacitet i veličinu populacije beskonačne, a zadana vrijednost za disciplinu raspoređivanja FCFS. Stoga M/M/1 red čekanja ima neograničenu količinu prostora za držanje kupaca koji čekaju, beskonačnu populaciju iz koje se izvlače kupci i primjenjuje FCFS politiku raspoređivanja (Sztrik, 2021:12-13).

5.2. Mreže čekanja

Do sada su razmatrani sustavi čekanja u redu u kojima svaki kupac dolazi u uslužni centar, biva uslužen od strane poslužitelja, a zatim odlazi i nikad se ne vrati. Ovo se ponekad naziva sustavom "jednog čvora". Sustav s "više čvorova" je onaj u kojem kupac zahtijeva uslugu na više čvorova. Takav sustav može se promatrati kao mreža čvorova, u kojoj je svaki čvor uslužni centar koji ima prostor za pohranu (međuspremnik) za formiranje redova čekanja i možda s više poslužitelja za obradu zahtjeva korisnika. Kroz ovo poglavlje koristit će se riječi "čvor" i "servisni/uslužni centar" naizmjenično. Korisnici ulaze u sustav dolaskom u jedan od uslužnih centara, stoje u redu i na kraju dobiju uslugu u tom centru, a po odlasku ili nastave do nekog drugog servisnog centra u mreži kako bi dobili dodatnu uslugu ili potpuno napuste mrežu.

Nije teško zamisliti primjere takvih mreža. Kada pacijenti dođu u liječničku ordinaciju, često trebaju ispuniti dokumentaciju, za potrebe osiguranja ili ažurirati svoju medicinsku dokumentaciju, a zatim odlaze do medicinske sestre radi raznih mjerenja kao što su težina, krvni tlak i slično. Sljedeća postaja obično je čekanje u redu (tj. strpljivo čekanje) da jedan od liječnika stigne i započne konzultacije i pregled. Možda će biti potrebno napraviti rendgenske snimke, ultrazvuk i tako dalje. Nakon što se ti postupci završe, možda će biti potrebno još jednom razgovarati s liječnikom. Krajnji centar kroz koji pacijent mora proći uvijek je naplatni ured. U svakoj od ovih različitih točaka, pacijent će možda morati čekati dok se drugi pacijenti liječe. Tako pacijent dolazi, prima uslugu/liječenje na više različitih točaka i konačno odlazi da se vrati na posao ili da ode kući (Bolch, 2006:321-325).

Brojna nova razmatranja javljaju se kada se promatraju mreže redova čekanja. Na primjer, topološka struktura mreže je važna jer opisuje dopuštene prijelaze između servisnih centara. Stoga bi bilo neprikladno da se pacijent udalji od medicinske sestre i ode izravno na rendgensko snimanje prije nego što se posavjetuje s liječnikom. Također, mora se nekako opisati put kojim prolaze pojedini kupci. U primjeru koji navodi Stewart (2014:559-560), neki pacijenti mogu napustiti mjesto medicinske sestre kako bi vidjeli dr. Sawbonesa, dok drugi mogu napustiti mjesto iste medicinske sestre kako bi vidjeli dr. Doolittlea. Neki od pacijenata dr. Sawbonesa možda će morati ići u rendgenski laboratorij, drugi na ultrazvuk, dok će treći možda biti poslani izravno na naplatu.

U mrežama čekanja potrebno je suočiti se sa situacijom u kojoj se korisnici koji odlaze iz jednog servisnog centra mogu pomiješati s korisnicima koji napuštaju drugi servisni centar, a kombinirani tok može biti predodređen za ulazak u treći servisni centar. Stoga postoji interakcija među redovima čekanja u mreži i to u konačnici može imati komplicirani učinak na proces dolaska u iduće servisne centre. Vrijeme dolaska korisnika može postati korelirano s vremenom pružanja usluga korisnicima nakon što korisnici prođu svoju točku ulaska u mrežu i ta korelacija može otežati matematičku analizu ovih redova čekanja (Stewart, 2014:560-562).

Sljedećim poglavljem započinje praktični dio ovog rada, čime će se zaokružiti cjelina o diskretnim simulacijama i redovima čekanja: od teorije do provedbe.

6. Postavka modela u Pythonu

Kako bi se uspješno simulirao red čekanja, valja postaviti model. Ovaj diplomski rad bavit će se diskretnom simulacijom reda čekanja na primjeru jedne poslovne banke.
Koristit će se programski jezik Python te biblioteke Ciw i Matplotlib. Sve simulacije korištene u ovom radu moguće je samostalno reproducirati kloniranjem repozitorija[2]. Kod je napisan prateći dokumentaciju i primjere spomenutih biblioteka. U nastavku će ukratko biti pojašnjeno za što služi pojedina biblioteka.

6.1. Ciw i Matplotlib

Prema službenoj Ciw dokumentaciji (2022), prevedeno s engleskog jezika Ciw je biblioteka simulacije diskretnih događaja za otvorene mreže čekanja. Njene temeljne značajke uključuju mogućnost simulacije mreža redova čekanja, više klasa korisnika i implementaciju blokiranja tipa I za ograničene mreže. Implementiran je i niz drugih značajki, uključujući prioritete, zaustavljanje, ponavljanje, rasporede, dolaske serija, dinamičke klase klijenata i otkrivanje zastoja.

Pomoću Ciw biblioteke moguće je izvršiti originalnu pretpostavljenu simulaciju, kao i "what-if" scenarije koji će biti detaljno objašnjeni u sljedećem poglavlju.

Riječima opisanim na službenoj stranici Matplotlib biblioteke (2022), ona je sveobuhvatna biblioteka za stvaranje statičnih, animiranih i interaktivnih vizualizacija u Pythonu.

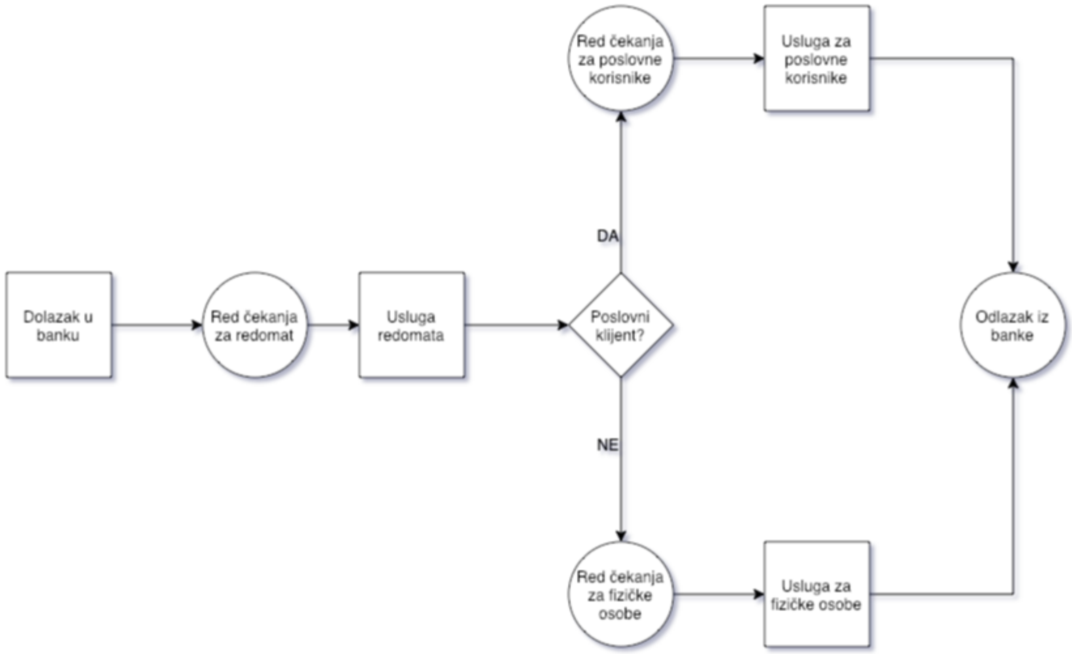
Pomoću Matplotliba moguće je vizualizirati dobivene rezultate kako bi ih se lakše moglo predočiti ciljanoj publici.

U nastavku će biti opisan primarni model simulacije.

[2] GitHub repozitorij (https://github.com/dominikandelic/disrektna-simulacija-redova-cekanja)

6.2. Model

Dijagram koji slijedi predstavlja temeljni model koji će se koristiti za simulaciju i analizu, kao i usporedbu s what-if scenarijima kojima će se nastojati poboljšati postojeći rezultati.



Slika 4. Model aktivnosti klijenata poslovne banke. Izvor: Izrada autora

Kao primjer uzet će se tipična poslovnica u nekoj poslovnoj banci.

Postavke su sljedeće:

- bazna vremenska jedinica su minute
- simulira se trajanje jedne smjene od 8 sati
- postoje dva tipa klijenata: poslovni korisnici i fizičke osobe (građanstvo)
- dolazak poslovnih korisnika je eksponencijalno distribuiran te iznosi 2 klijenta po 60 minuta (satu)
- dolazak fizičkih osoba (građanstva) je eksponencijalno distribuiran te iznosi 12 klijenata po 60 minuta
- postoje 3 čvora: redomat (pametni dispenzer papirića koji ispisuju poziciju u redu čekanja) – Čvor 1, usluživanje fizičkih osoba – Čvor 2 te usluživanje poslovnih korisnika – Čvor 3
- vrijeme usluge (obrade) poslovnih korisnika je eksponencijalno distribuirano te iznosi 1 klijent svakih 30 minuta
- vrijeme usluge (obrade) fizičkih osoba je eksponencijalno distribuirano te iznosi 1 klijent svakih 15 minuta

- dodijeljen je 1 poslužitelj za čvor recepcije, 2 poslužitelja za čvor usluživanja fizičkih osoba i 1 poslužitelj za čvor usluživanja poslovnih korisnika
- obje klase klijenata dolaze pred redomat te trajanje usluge redomata (biranje željene usluge i izdavanje papirića) iznosi 1 klijent po minuti
- nakon što dobiju papirić, klijenti ovisno o svojoj klasi čekaju u tom redu za traženu uslugu

U nastavku slijedi slika koja prikazuje kako to izgleda zapisano u Pythonu.

```
import ciw
import matplotlib.pyplot as plt

N = ciw.create_network(
    arrival_distributions={'Class 0': [ciw.dists.Exponential(rate=0.033),
                                       ciw.dists.NoArrivals(),
                                       ciw.dists.NoArrivals()],
                           'Class 1': [ciw.dists.Exponential(rate=0.2),
                                       ciw.dists.NoArrivals(),
                                       ciw.dists.NoArrivals()]},
    service_distributions={'Class 0': [ciw.dists.Exponential(rate=1.0),
                                       ciw.dists.Deterministic(value=0.0),
                                       ciw.dists.Exponential(rate=0.033)],
                           'Class 1': [ciw.dists.Exponential(rate=1.0),
                                       ciw.dists.Exponential(rate=0.067),
                                       ciw.dists.Deterministic(value=0.0)]},
    routing={'Class 0': [[0.0, 0.0, 1.0],
                         [0.0, 0.0, 0.0],
                         [0.0, 0.0, 0.0]],
             'Class 1': [[0.0, 1.0, 0.0],
                         [0.0, 0.0, 0.0],
                         [0.0, 0.0, 0.0]]},
    number_of_servers=[1, 2, 1],
)
```

Slika 5. Postavka modela u Pythonu. Izvor: Izrada autora

U Pythonu *Class 0* predstavljen je poslovnim korisnikom, a *Class 1* fizičkom osobom.

Parametar *arrival_distributions* označava distribuciju dolazaka, a *service_distributions* distribuciju usluživanja.

Routing predstavlja $n * n$ matricu za svaku klasu klijenta koja definira smjer kretanja iz i-tog čvora u j-oti čvor.

Primjerice, za poslovne korisnike 1. red znači da će 100% usluženih na 1. čvoru otići u 3. stupac, odnosno u 3. čvor, za sve ostale redove je prazno, tj. 0%.

Zapisano u Kendallovoj notaciji:

1. Čvor – M/M/1
2. Čvor – M/M/2
3. Čvor – M/M/1

Sada kada je definiran i postavljen model u Pythonu, potrebno je izvršiti simulaciju kako bi se prikupili potrebni rezultati za analizu i postavljanje what-if scenarija.

7. Rezultati simulacije

Kako bi rezultati bili vjerodostojniji, uzet će se veći broj iteracija simulacije, u ovom slučaju radit će se s 500 iteracija i bilježiti rezultati. Slijedi prikaz kako to izgleda zapisano u Pythonu.

```
average_waits_business = []
average_waits_civilian = []

for trial in range(500):
    ciw.seed(trial)
    Q = ciw.Simulation(N)
    Q.simulate_until_max_time(480)
    recs = Q.get_all_records()
    waits_business = [r.waiting_time for r in recs if r.customer_class == 0]
    waits_civilian = [r.waiting_time for r in recs if r.customer_class == 1]
    mean_wait_business = sum(waits_business) / len(waits_business)
    mean_wait_civilian = sum(waits_civilian) / len(waits_civilian)
    average_waits_business.append(mean_wait_business)
    average_waits_civilian.append(mean_wait_civilian)


average_mean_wait_business = sum(average_waits_business) / len(average_waits_business)
average_mean_wait_civilian = sum(average_waits_civilian) / len(average_waits_civilian)

fig, ax = plt.subplots()
ax.set_title('Distribucija prosječnih vremena čekanja (u 500 iteracija)')
ax.set_xlabel("Vrijeme čekanja (min)")
ax.set_ylabel("Frekvencija pojavljivanja (u broju iteracija)")

ax.hist(average_waits_business, label="Prosječno vrijeme čekanja za poslovne klijente", alpha = 0.5)
ax.hist(average_waits_civilian, label="Prosječno vrijeme čekanja za poslovne klijente", alpha = 0.5)
ax.legend()

plt.show()

print("Prosječno vrijeme čekanja za poslovne korisnike: " + str(round(average_mean_wait_business, 2)) + "minuta")
print("Prosječno vrijeme čekanja za građanstvo: " + str(round(average_mean_wait_civilian, 2)) + "minuta")
```

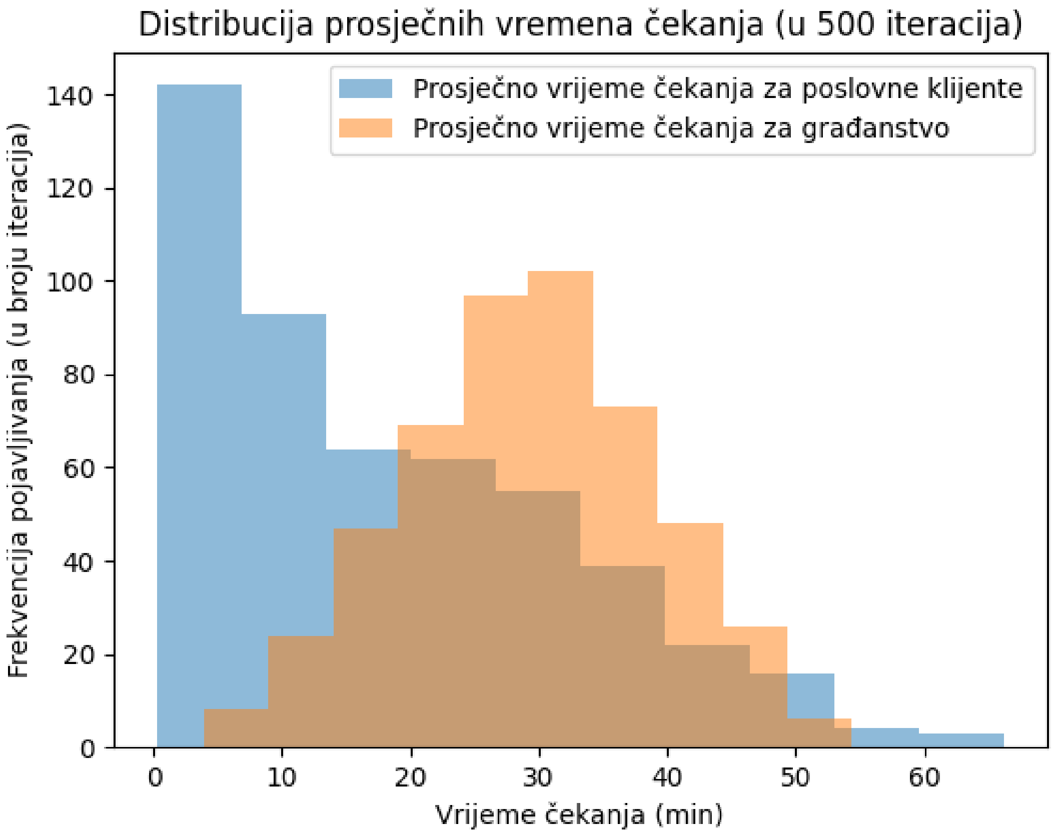
Slika 6. Prikaz 500 iteracija simulacije u Pythonu. Izvor: Izrada autora

Važno je za napomenuti da metoda *seed* postavlja „sjeme" kojim se generiraju pseudo-slučajne vrijednosti, te je u svakoj analizi koja slijedi rezultat uvijek jednak budući da se koristi raspon od 0 do 499 prilikom svakog pokretanja simulacije.

Pokretanjem simulacije iznad dobili su se sljedeći rezultati:

- Prosječno vrijeme čekanja za poslovne korisnike: 18.31minuta
- Prosječno vrijeme čekanja za građanstvo: 29.25minuta

Kako bi se mogla bolje predočiti distribucija vremena čekanja, korištena je biblioteka Matplotlib te je u nastavku prikaz distribucije prosječnih vremena čekanja za poslovne korisnike i fizičke osobe.



Slika 7. Distribucija prosječnih vremena čekanja (u 500 iteracija). Izvor: Izrada autora

Iz histograma se može vidjeti kako poprilično velik broj puta je vrijeme čekanja za poslovne klijente iznosilo od 0 do 10 minuta, što i nije toliko velik broj.
U kontrastu, vrijeme čekanja za fizičke osobe najviše se kretalo u rasponu od 20-40 minuta, što odgovara i dobivenom prosjeku.

Treba obratiti pažnju i na broj klijenata koji su ušli i oni koji su uspješno završili s uslugom u cijelom sustavu. Rezultati su sljedeći:

- Prosječno ušlo klijenata u sustav: 112
- Prosječno izašlo klijenata iz sustava: 74

Brojke ukazuju na to da, na kraju radnog vremena od 8h, čak 38 klijenata je ostalo unutar sustava.

Sljedeći kratki kod prikazuje način mjerenja najdužeg vremena čekanja entiteta u procesu.

```python
max_waiting_times = []

for trial in range(500):
    ciw.seed(trial)
    Q = ciw.Simulation(N)
    Q.simulate_until_max_time(480)
    recs = Q.get_all_records()
    max_waiting_time_per_iteration = max([r.waiting_time for r in recs])
    max_waiting_times.append(max_waiting_time_per_iteration)


print("Najduže vrijeme čekanja: " + str(round(max(max_waiting_times), 2)) + "min")
```

Slika 8. Izračun najdužeg vremena čekanja entiteta u procesu. Izvor: Izrada autora

Pokretanjem navedenog dijela koda, dobije se sljedeće:

- Najduže vrijeme čekanja: 366.8min

U nastavku slijedi prikaz koda koji se bavi računanjem prosječne iskorištenosti čvorova.

```python
average_utilization_node_1 = []
average_utilization_node_2 = []
average_utilization_node_3 = []

for trial in range(500):
    ciw.seed(trial)
    Q = ciw.Simulation(N)
    Q.simulate_until_max_time(480)
    recs = Q.get_all_records()
    node_1_queue_utilization = Q.transitive_nodes[0].server_utilisation
    node_2_queue_utilization = Q.transitive_nodes[1].server_utilisation
    node_3_queue_utilization = Q.transitive_nodes[2].server_utilisation
    average_utilization_node_1.append(node_1_queue_utilization)
    average_utilization_node_2.append(node_2_queue_utilization)
    average_utilization_node_3.append(node_3_queue_utilization)

average_mean_utilization_node_1 = sum(average_utilization_node_1) / len(average_utilization_node_1)
average_mean_utilization_node_2 = sum(average_utilization_node_2) / len(average_utilization_node_2)
average_mean_utilization_node_3 = sum(average_utilization_node_3) / len(average_utilization_node_3)

fig, ax = plt.subplots()
ax.set_title('Distribucija iskorištenosti čvorova (u 500 iteracija)')
ax.set_xlabel("Iskorištenost čvora")
ax.set_ylabel("Frekvencija pojavljivanja (u broju iteracija)")

ax.hist(average_utilization_node_1, label="Prosječna iskorištenost čvora 1", alpha = 0.5)
ax.hist(average_utilization_node_2, label="Prosječna iskorištenost čvora 2", alpha = 0.5)
ax.hist(average_utilization_node_3, label="Prosječna iskorištenost čvora 3", alpha = 0.5)
ax.legend()

plt.show()

print("Prosječna iskorištenost čvora 1: " + str(round(average_mean_utilization_node_1, 4) * 100) + "%")
print("Prosječna iskorištenost čvora 2: " + str(round(average_mean_utilization_node_2, 4) * 100) + "%")
print("Prosječna iskorištenost čvora 3: " + str(round(average_mean_utilization_node_3, 4) * 100) + "%")
```

Slika 9. Prosječna iskorištenost čvorova u 500 iteracija. Izvor: Izrada autora

Valja se podsjetiti na raspored čvorova:

1. Čvor – redomat
2. Čvor - usluživanje fizičkih osoba
3. Čvor – usluživanje poslovnih korisnika

Pokretanjem ove simulacije dobiju se sljedeći rezultati:

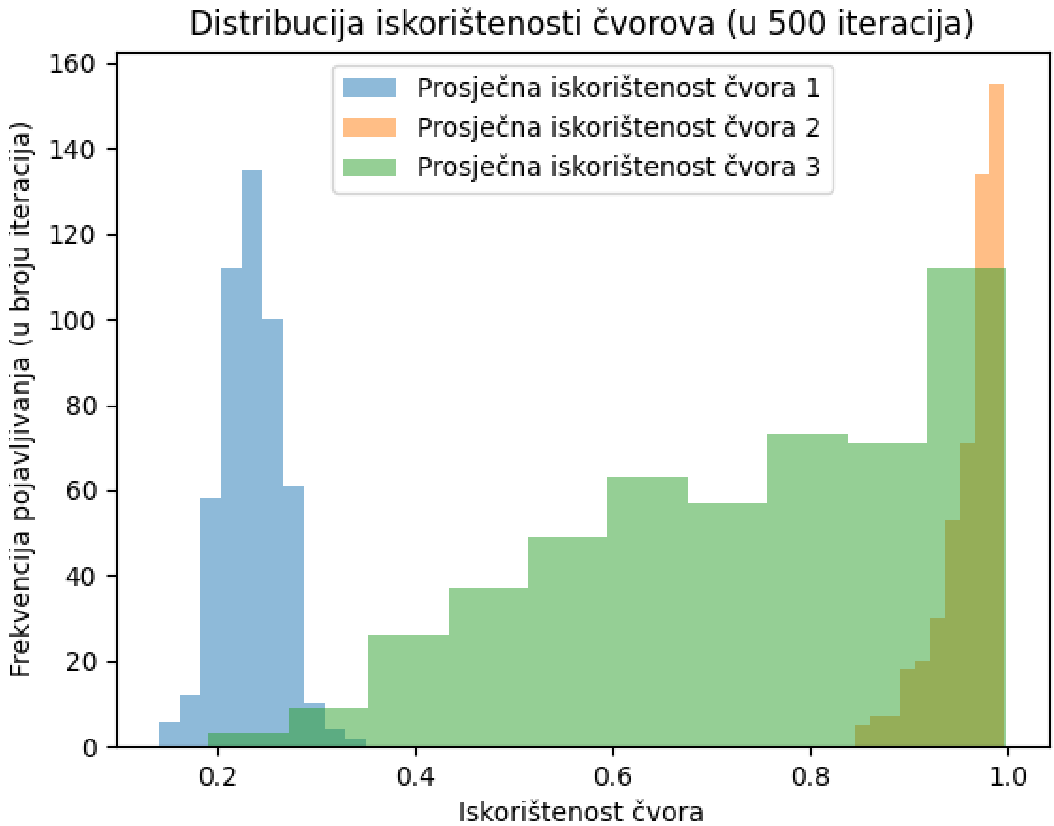
- Prosječna iskorištenost čvora 1: 23.39%
- Prosječna iskorištenost čvora 2: 96.12%
- Prosječna iskorištenost čvora 3: 73.69%

Sasvim je logično za očekivati da redomat nije niti približno u potpunosti iskorišten, tek 23.39%, budući da je njegova funkcija jednostavna i kratka.

Prosječna iskorištenost usluge poslovanja s fizičkim osobama, odnosno sektorom građanstva je 96.12%, označavajući visoku iskorištenost.

Prosječna iskorištenost usluge poslovanja s poslovnim korisnicima je 73.69%.

U nastavku slijedi histogram koji pokazuje distribuciju iskorištenosti čvorova.



Slika 10. Distribucija iskorištenosti čvorova (u 500 iteracija). Izvor: Izrada autora

Jasno je za zaključiti kako se distribucija čvora 1 kreće oko prosjeka. Čvor 2 u više od 150 ponavljanja ima 100% iskorištenost, što znači da je taj čvor preopterećen.

Čvor 3, odnosno usluživanje poslovnih korisnika ima dosta raspršenu iskorištenost.

Potrebno je napraviti malo opsežniju analizu; u nastavku će se izračunati standardna devijacija i medijan za sva tri čvora, s naglaskom na treći čvor kako bi se došlo do boljih zaključaka, budući da je on jedini koji nema „jasnu“ sliku.

Metode u Pythonu za računanje standardne devijacije i medijana dostupne su unutar biblioteke *statistics*.

Budući da je sve ostalo isto, u nastavku će biti prikazan samo dio koda koji se tiče izračuna standardne devijacije i medijana.

```
print("Standardna devijacija čvora 1: " + str(round(stdev(average_utilization_node_1), 4) * 100) + "%")
print("Standardna devijacija čvora 2: " + str(round(stdev(average_utilization_node_2), 4) * 100) + "%")
print("Standardna devijacija čvora 3: " + str(round(stdev(average_utilization_node_3), 4) * 100) + "%")


print("Medijan iskorištenosti čvora 1: " + str(round(median(average_utilization_node_1), 4) * 100) + "%")
print("Medijan iskorištenosti čvora 2: " + str(round(median(average_utilization_node_2), 4) * 100) + "%")
print("Medijan iskorištenosti čvora 3: " + str(round(median(average_utilization_node_3), 4) * 100) + "%")
```

Slika 11. Prikaz statističkih metoda za izračun medijana i standardne devijacije. Izvor: Izrada autora

Valja se prisjetiti da su rezultati uvijek isti budući da je *seed* postavljen kao nepromjenjiv, odnosno u simulaciji od 500 iteracija *seed* je u rasponu od 0 do 499.

Pokretanjem navedenog koda, dobiju se sljedeći rezultati:

- Standardna devijacija čvora 1: 3.11%
- Standardna devijacija čvora 2: 3.05%
- Standardna devijacija čvora 3: 19.06%
- Medijan iskorištenosti čvora 1: 23.33%
- Medijan iskorištenosti čvora 2: 97.04%
- Medijan iskorištenosti čvora 3: 76.51%

Za čvor 3 može se zaključiti kako u 50% iteracija iskorištenost je bila manja od 76.51%, a u drugih 50% veća od 76.51%. To se dodatno može potvrditi gledajući standardnu devijaciju koja pokazuje kako prosječna vrijednost pretjerano ne vrijedi za čvor 3, budući da je standardno odstupanje od prosjeka čak 19.06%.

Uzimajući sve u obzir, dobila se jasnija slika o stanju iskorištenosti čvorova. Sada je na redu provođenje what-if scenarija kojim će se nastojati optimizirati sustav.

8. What-if analize

Proučivši izvješće simulacije može se donijeti nekoliko bitnih tvrdnji.

Čvor redomata iskorišten je tek 23.39%, no budući da je to resurs koji i nema neku posebnu uslugu osim izdavanja papirića za čekanje u redu ne može ga se optimizirati, tj. dodavanje još jednog redomata ne bi imalo smisla.
Područje u kojem sustav „zapinje" jest čvor 2, odnosno usluživanje sektora građanstva. Njegova iskorištenost u više od 150 iteracija iznosi 100%, što znači da je sustav preopterećen. Budući da je u sustavu tijekom osmosatnog radnog vremena ostalo čak 38 klijenata, kada bi došlo do nekakve navale u pogledu povećanja broja klijenata, već zagušeni sustav bi potpuno „izgubio konce".
Nešto što je zanimljivo za proučiti je čvor 3 (sektor poslovanja s poslovnim korisnicima). Njegova iskorištenost poprilično varira te će predmet jedne od what-if analiza biti dodjeljivanje tog čvora fizičkim osobama kada taj resurs nije zauzet poslovnim korisnicima.

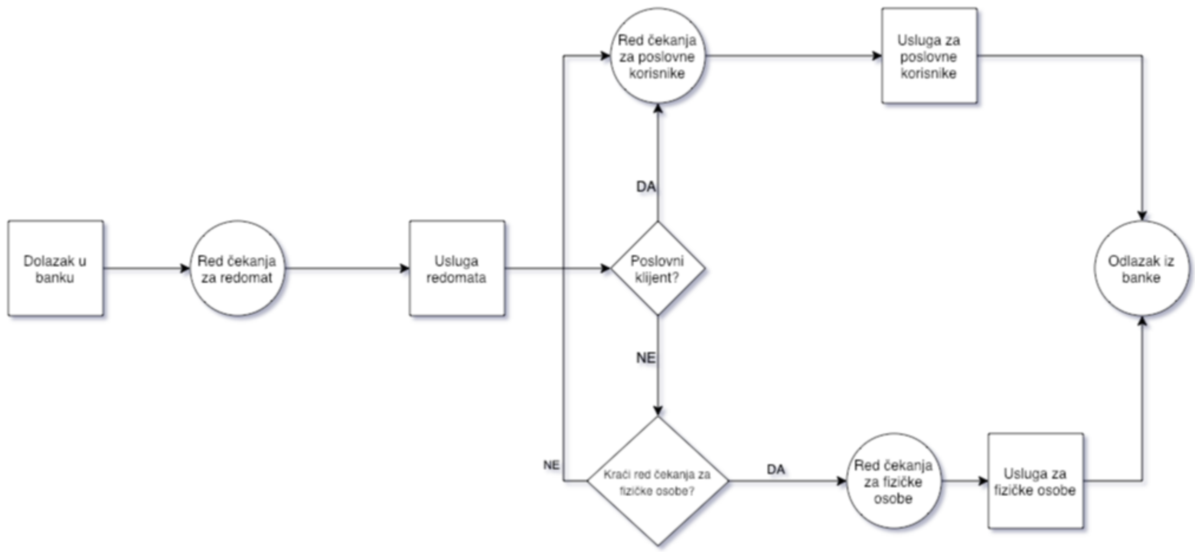
Dakle, rad će promatrati 2 what-if analize te sukladno rezultatima dati preporuke treba li poslovanje razmotriti ponuđene opcije ili ih odbaciti.

8.1. Prvi scenarij: Raspoređivanje po zauzetosti

U prvom scenariju logika je sljedeća: poslovni korisnici će kao i u originalnom modelu s čvora 1 (redomata) biti preusmjereni na čvor 3. Ako postoji netko ispred njih, stoje u redu za čvor 3. No, promjena je kod sektora građanstva.

Budući da se ustanovilo kako je taj čvor preopterećen, postavlja se sljedeće pitanje: sustav koji pokreće redomat zna koliko ljudi čeka na koju uslugu; na osnovu toga mogao bi se preusmjeriti sektor građanstva na onaj čvor koji je manje zauzet (čvor 2 ili 3). Upravo to je i napravljeno u ovom scenariju.

Potrebno je izmijeniti originalni model budući da dolazi do dodatne provjere, odnosno raspoređivanja prilikom usmjeravanja s čvora redomata na idući čvor. Slijedi slika izmijenjenog modela.



Slika 12. Prikaz izmijenjenog modela s dodatnim uvjetom raspoređivanja. Izvor: Izrada autora

Može se vidjeti da je dodan uvjet koji provjerava je li kraći red čekanja za fizičke osobe te ako jeste, usmjerava klijenta koji je fizička osoba u taj red. U suprotnom će se klijent koji pripada sektoru građanstva preusmjeriti u red čekanja za poslovne korisnike.

U nastavku slijedi kod koji prikazuje kako je to zapisano u *Pythonu* pomoću *Ciwa.*

```
class CustomRouting(ciw.Node):
    def next_node(self, ind):
        if ind.customer_class == 1:
            n2 = self.simulation.nodes[2].number_of_individuals
            n3 = self.simulation.nodes[3].number_of_individuals
            if n2 < n3:
                return self.simulation.nodes[2]
            elif n3 < n2:
                return self.simulation.nodes[3]
            return ciw.random_choice([self.simulation.nodes[2], self.simulation.nodes[3]])
        else:
            return self.simulation.nodes[3]
```

Slika 13. Prikaz logike raspoređivanja po zauzetosti. Izvor: Izrada autora

Dakle, logika je poprilično jednostavna: nakon što redomat završi uslugu, preusmjerit će korisnika ovisno o njegovoj „klasi“, odnosno o tipu korisnika.

Klasa 1 predstavlja sektor građanstva, a klasa 0 poslovne korisnike. Ako je klijent poslovni korisnik, bit će preusmjeren na čvor 3, odnosno na sektor usluživanja za poslovne korisnike. Ako je klijent fizička osoba, ovisno o zauzetosti čvorova 2 i 3 bit će preusmjeren u onaj manje zauzet.

Ostatak modela je jednak, jedino što se promijenilo jest strategija preusmjerivanja koja sada ovisi o zauzetosti čvorova (za fizičke osobe). Valjalo bi sada pokrenuti simulaciju s 500 iteracija i promotriti rezultate.

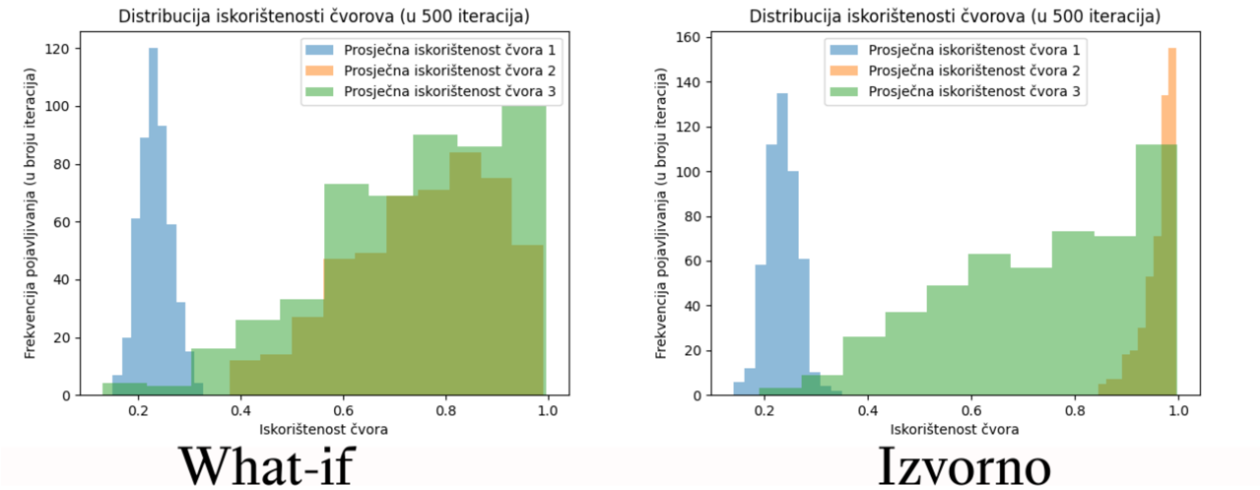
Kao što je napomenuto, kod za generiranje izvješća je identičan kao prethodno već spomenuti. Pokrenuvši program, dobiju se sljedeći rezultati:

- Prosječno ušlo klijenata u sustav: 112
- Prosječno izašlo klijenata iz sustava: 97
- Prosječna iskorištenost čvora 1: 23.38%
- Prosječna iskorištenost čvora 2: 75.61%
- Prosječna iskorištenost čvora 3: 73.77%
- Prosječno vrijeme čekanja za poslovne korisnike: 17.83minuta
- Prosječno vrijeme čekanja za građanstvo: 11.1minuta
- Najduže vrijeme čekanja: 359.94min

Odmah se može vidjeti kako je u usporedbi s početnom simulacijom ogromna razlika u broju klijenata koji su ostali u sustavu nakon osmosatnog radnog vremena.
U ovom scenariju ostalo je 15 klijenata u sustavu, u usporedbi s 38 u izvornoj simulaciji, što je čak 60% manje, vrlo zadovoljavajući rezultat.

Nadalje, ako se promatra iskorištenost čvorova 2 i 3 također se mogu vidjeti razlike. Najbolje će se uočiti razlike usporedbom distribucija iskorištenosti čvorova između what-if analize i originalne postavke simulacije.

U nastavku slijedi slika usporedbe gdje se jasno može vidjeti razlika u iskorištenosti.



Slika 14. Usporedba distribucija iskorištenosti (u 500 iteracija). Izvor: Izrada autora

Može se zaključiti kako se, u odnosu na originalnu postavku, promijenila distribucija iskorištenosti. Čvor 2 više nije preopterećen te mu je distribucija sličnija, ali ne ista, distribuciji čvora 3. Budući da su čvorovi 2 i 3 sada na oko 75% iskorištenosti, može se reći da su optimalno iskorišteni.

Isto tako, drastična razlika se dogodila unutar kategorije prosječnog vremena čekanja: s izvornih prosječnih 29.25minuta za građanstvo, sada je taj prosjek 11.1minutu, što je smanjenje od čak 62%. Što se tiče poslovnih korisnika, njihovo prosječno vrijeme čekanja se nije drastično promijenilo.

Naposlijetku, najduže vrijeme čekanja se isto tako smanjilo, ali samo za ~6.8 minuta, što i nije neko veliko smanjenje.

8.2. Drugi scenarij: Povećanje resursa

Ovaj scenarij je jednostavniji za implementirati, ali je ujedno i skuplji. Pretpostavka je sljedeća: ako je čvor 2 (usluživanje sektora građanstva) s 2 poslužitelja u prosjeku na 97%-tnoj iskorištenosti, pametno bi bili zaposliti dodatne resurse.

Provest će se simulacija i analiza 2 slučaja:

1. Zapošljavanje dodatnog bankara na čvoru 2
2. Zapošljavanje dva dodatna bankara na čvoru 2

8.2.1. Zapošljavanje jednog dodatnog bankara

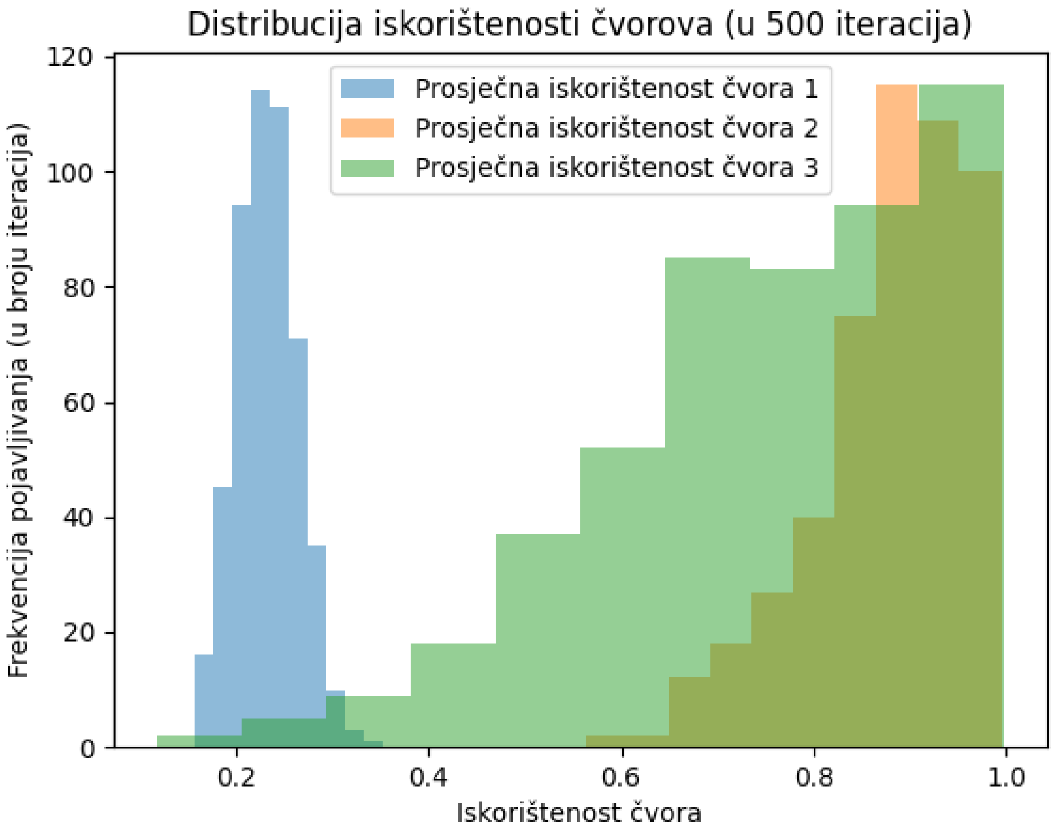
Što se modela tiče, sve ostaje nepromijenjeno osim parametra *number_of_servers* koji sada ima 3 poslužitelja na čvoru 2 u odnosu na originalnu postavku gdje su zaposlena 2 poslužitelja. Čvor 2 zapisan u Kendallovoj notaciji je sada M/M/3.

Pokretanjem 500 iteracija ove simulacije dobiju se sljedeći rezultati:

- Najduže vrijeme čekanja: 315.29min
- Prosječno ušlo klijenata u sustav: 113
- Prosječno izašlo klijenata iz sustava: 97
- Prosječna iskorištenost čvora 1: 23.35%
- Prosječna iskorištenost čvora 2: 87.99%
- Prosječna iskorištenost čvora 3: 75.69%
- Prosječno vrijeme čekanja za poslovne korisnike: 18.74minuta
- Prosječno vrijeme čekanja za građanstvo: 11.33minuta

Zanimljivo je kako se najduže vrijeme čekanja smanjilo za 14%. Nadalje, 1 klijent više je u prosjeku ušao u sustav u odnosu na početnu simulaciju.

Prosječna iskorištenost čvora 2 više nije 97% te sada iznosi 88%, što je smanjenje za ~9%. U nastavku slijedi prikaz distribucije iskorištenosti čvorova za ovaj *what-if* scenarij.



Slika 15. Distribucija iskorištenosti čvorova (u 500 iteracija) sa zapošljavanjem dodatnog bankara na čvoru 2.

Izvor: Izrada autora

Iz histograma se može vidjeti kako je iskorištenost opet sličnog izgleda (poput nekog tornja), gdje su ponavljanja najviše koncentrirana na oko 90% iskorištenosti. Ovo signalizira malo standardno odstupanje.

Kada se promatraju prosječna vremena čekanja, može se vidjeti smanjenje u vremenu čekanja za sektor građanstva – s 29.25 minuta na tek 11.33 minute, što je smanjenje od 61%.

Valjalo bi sada promotriti i drugi slučaj.

8.2.2. Zapošljavanje dva dodatna bankara

U ovom modelu, kao i u prvom slučaju, sve je ostalo nepromijenjeno osim *number_of_servers* parametra koji je sada postavljen na 4 poslužitelja za čvor 2, odnosno u Kendallovoj notaciji M/M/4.

Kada se pokrene simulacija od 500 iteracija osmosatnog radnog dana, dobiju se sljedeći rezultati:

- Najduže vrijeme čekanja: 334.73min
- Prosječno ušlo klijenata u sustav: 112
- Prosječno izašlo klijenata iz sustava: 103
- Prosječna iskorištenost čvora 1: 23.29%
- Prosječna iskorištenost čvora 2: 71.14%
- Prosječna iskorištenost čvora 3: 76.21%
- Prosječno vrijeme čekanja za poslovne korisnike: 19.19minuta
- Prosječno vrijeme čekanja za građanstvo: 2.88minuta

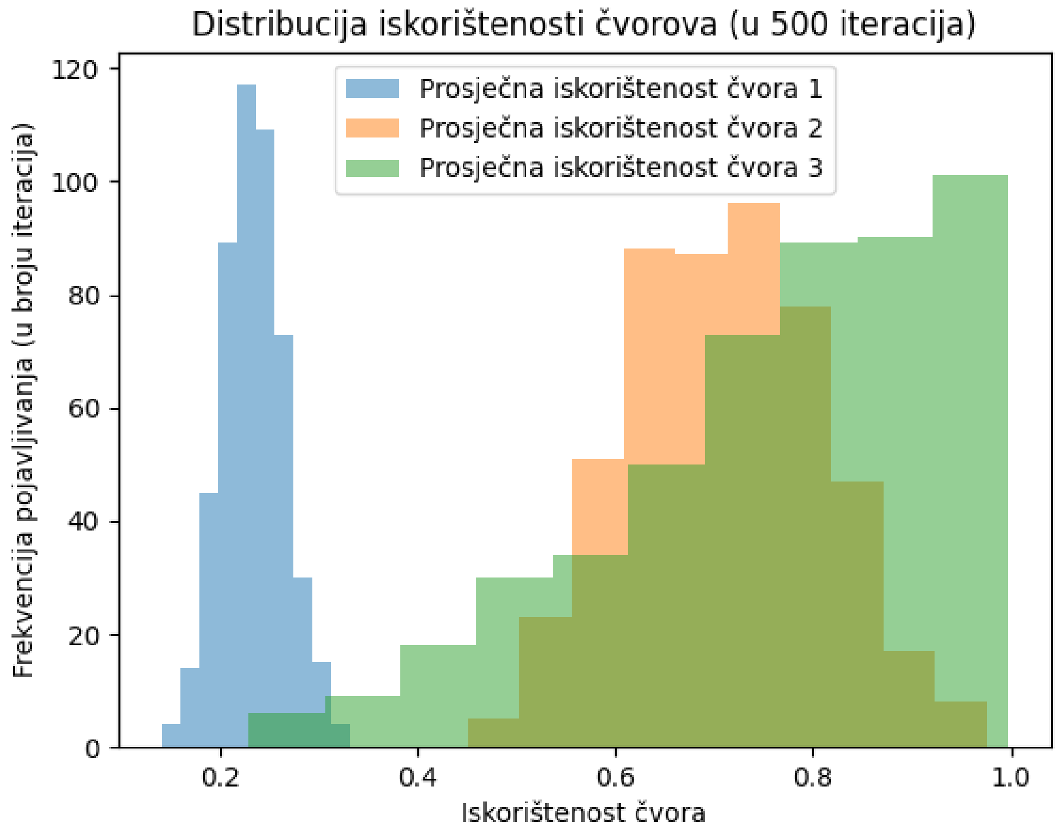
Kada se uspoređuje s izvornom simulacijom, broj klijenata koji su ušli u sustav je ostao nepromijenjen, no broj klijenata koji su izašli iz sustava sada iznosi 103, što je povećanje za 39%. U usporedbi s ostalim scenarijima i slučajevima, ovo je najbolji rezultat u ovoj kategoriji.

Ako se pogleda prosječno vrijeme čekanja, dogodila se nepovoljna situacija u ovom slučaju: prosječno vrijeme čekanja za poslovne korisnike se povećalo. No s druge strane, vrijeme čekanja za građanstvo je sada gotovo nepostojeće, tek 2.88minuta.

Najduže vrijeme čekanja se smanjilo za ~8%, što je pozitivno.

Posljednje, promotrit će se prosječna iskorištenost čvorova. Čvor 1 je ostao gotovo nepromijenjen što je i očekivano. Čvor 2 je sada 71.14% prosječno iskorišten, što je u odnosu na originalnu postavku smanjenje od čak 26%. Prosječna iskorištenost čvora 3 povećala se za 3%.

Slijedi prikaz distribucije iskorištenosti čvorova u ovoj simulaciji.



Slika 16. Prikaz distribucije iskorištenosti čvorova (u 500 iteracija) zapošljavanjem dva dodatna bankara na čvoru 2. Izvor: Izrada autora

Kao što se može vidjeti, iskorištenost čvora 2 više nije u obliku „tornja“ koncentriranog oko 100% iskorištenosti, već je sada šira distribucija od ~55% do ~85%. Zaključuje se da je sada iskorištenost optimalna i da se ne stavlja toliki stres na čvor 2.

8.2.3. Usporedba slučajeva

Stavivši oba slučaja usporedno, može se zaključiti kako je puno učinkovitije zaposliti dva dodatna bankara, budući da je više klijenata izišlo iz sustava. Isto tako, prosječno čekanje sektora građanstva svedeno je na minimum. Samim time, taj sustav podržava efikasniju uslugu zbog smanjenja vremena čekanja kao i veću količinu „obrađenih“ klijenata.

Međutim, usporedbu ova dva slučaja treba promatrati i iz troškovne perspektive. Zapošljavanje dodatnog bankara iziskuje dodatan mjesečni trošak. Zapošljavanjem dva dodatna bankara, troškovi su još veći.

Kada bi se pretpostavilo da banka ima iznimno pozitivne neto gotovinske tokove, tada bi definitivno prevagnuo slučaj zapošljavanja dva bankara. No, ako se pretpostavi nekakav srednji slučaj, gdje je stanje povoljno, ali ne i toliko optimistično, ipak bi se trebalo odvagnuti za jeftiniju opciju, budući da ona smanjuje prosječno vrijeme čekanja za čak 62%.

8.3. Zaključak svih scenarija

Sada kada je provedena simulacija sva 3 slučaja, valjalo bi dati konkretni zaključak.

Kada bi se zanemarilo da zapošljavanje dodatnog osoblja iziskuje mjesečne troškove, prevagnuo bi slučaj zapošljavanja 2 dodatna bankara za čvor usluživanja građanstva te bi to bila primarna preporuka imaginarnim naručiteljima ovih analiza.

No, kako u praksi ta situacija i nije baš realna, najizglednija opcija za prihvatiti jest ona prva provedena, odnosno korištenje strategija raspoređivanja po zauzetosti. Tom strategijom bi oba čvora bila optimalno iskorištena te bi se smanjilo vrijeme čekanja za građanstvo, kao i povećao broj klijenata koji su završili sa svim uslugama i izašli iz sustava. Isto tako, ne bi bilo potrebe zaposliti dodatne bankare te bi se, zbog povećanja broja „obrađenih" korisnika povećali prihodi banke, kao i stopa zadovoljstva korisnika zbog kraćeg vremena čekanja.

Primarna preporuka (novac nije presudan):
Provođenje strategije zapošljavanja 2 dodatna bankara za čvor 2.

Sekundarna preporuka (novci igraju ulogu, ali je ipak potrebno nešto učiniti):
Provođenje strategije raspoređivanja fizičkih osoba na osnovu zauzetosti čvorova 2 i 3.

9. Zaključak

Tema ovog diplomskog rada bila je diskretna simulacija redova čekanja. Glavni cilj rada bio je postaviti teorijsku podlogu diskretnih simulacija i redova čekanja, a zatim na primjeru pokazati i dokazati da simulacije imaju zaista moćnu primjenu u analizi i projektiranju za „stvarni svijet“. Simulacijski modeli se „pokreću“, ne rješavaju te se generira umjetna povijest sustava u kojem se atributi mijenjaju ovisno o događajima. Simulacija ima prednosti i nedostatke, ali je iznimno korisna pri dizajniranju novog sustava ili optimiziranju postojećeg. Brojna su područja primjene diskretnih simulacija: proizvodni sustavi, javni sustavi, transportni, uslužni, itd.

Prilikom izgradnje simulacijskih modela, bitno je imati minimalnu razinu detalja koji su kompatibilni s odgovorima iz simulacije. Preporučljivo je krenuti od osnovnih elemenata te njih postupno usavršavati.

Glavni elementi diskretne simulacije su entiteti i operacije. Entiteti se dijele na stalne i privremene, te aktivne i pasivne. Stalni entiteti su primjerice zaposlenik na šalteru banke, blagajnica u trgovini i slično, a privremeni entitetu su klijenti, kupci i slično; oni koji samo prolaze kroz sustav. Operacije predstavljaju dinamiku sustava te se dijele na događaje, procese i aktivnosti. Događaji su trenutci koji uzrokuju promjenu stanja u sustavu. Aktivnost je radnja koja se odvija između dva događaja i u njoj sudjeluju pasivni i aktivni entitet, a procesi su nizovi aktivnosti.

Teorija redova zanimljiv je fenomen, budući da se ljudi svakodnevno susreću s redovima čekanja. Zbog neizvjesnosti, mnoge svakodnevne situacije koje iziskuju neku potražnju za uslugom podložne su stvaranju redova budući da oni koji pružaju te usluge ne mogu odmah zadovoljiti zahtjeve. Primjerice, klijenti neke banke koji čekaju svoj red, automobili na semaforima, ljudi u IKEA-inom samoposlužnom restoranu i slične situacije su slučajevi u kojima su primjenjive teorije redova čekanja.

Red čekanja predstavlja sustav nepraznog skupa poslužitelja te isti pružaju usluge korisnicima koji ako čekaju na usluge stoje u redu čekanja u međuspremniku.

Redoslijed posluživanja korisnika ovisi o odabranoj uslužnoj disciplini. Zbog nasumičnih dolazaka kupaca nastaju redovi i kada raspoloživi kapacitet sustava nije veći od broja dolazaka. Kako bi uspješno karakterizirali redove čekanja, potrebno je koristiti Kendallovu notaciju.

Kada je kupcu potrebna usluga na više čvorova unutar istog sustava, tada se koriste mreže čekanja, gdje svaki čvor predstavlja uslužni centar sa svojim međuspremnikom za redove čekanja. Taj sustav može imati više poslužitelja za obradu zahtjeva korisnika.

Obradom teorijskog dijela rada, praktični dio bavio se simulacijom modela jedne poslovne banke. Usluživanje sektora građanstva trajalo je poprilično dugo te je bilo potrebno to detaljnije istražiti. Na temelju 500 iteracija, došlo se do kvalitetnih činjenica te su slijedile what-if analize, odnosno scenariji. Od ukupno 3 what-if scenarija, postojao je najbolji koji je sveo prosječno vrijeme čekanja za sektor građanstva na minimalno vrijeme te se imaginarnim naručiteljima analize taj prijedlog postavio kao primarni. No, problem je bio što je tražio dodatne mjesečne troškove u vidu zapošljavanja novih bankara. Samim time, postavio se i sekundarni prijedlog: raspoređivanje fizičkih osoba na temelju zauzetosti čvorova.

Literatura

1. Anđelić, D., 2022. *Diskretna simulacija redova čekanja - repozitorij praktičnog dijela.* [Mrežno] Available at: https://github.com/dominikandelic/disrektna-simulacija-redova-cekanja [Pokušaj pristupa 29. kolovoza 2022.].
2. Banks, J., Carson II, J. S., Nelson, B. L. & Nicol, D. M., 2010. *Discrete-Event System Simulation.* Upper Saddle River: Pearson.
3. Barković, D., 2001. *Operacijska istraživanja.* Osijek: Ekonomski fakultet.
4. Barković, D., 2011. *Uvod u operacijski menadžment.* Osijek: Ekonomski fakultet.
5. Bolch, G., Greiner, S., de Meer, H. & Trivedi, K., 2006. *Queueing Networks and Markov Chains: Modeling and Performance Evalutation with Computer Science Applications.* Hoboken: Wiley.
6. Božikov, J., 2007. Modeliranje i simulacija. U: J. Kern, ur. *Medicinskoinformatičke metode.* Zagreb: Medicinski fakultet Sveučilišta u Zagrebu i Medicinska Naklada , pp. 85-100.
7. Ciw, 2022. *Welcome to Ciw's documentation! — Ciw 2.3.1 documentation.* [Mrežno] Available at: https://ciw.readthedocs.io/en/latest/ [Pokušaj pristupa 21. kolovoza 2022.].
8. Collins, A. J., Sabz Ali Pour, F. & Jordan, C. A., 2021. Past challenges and the future of discrete event simulation. *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology,* pp. 1-19.
9. Čerić, V., 1993. *Simulacijsko modeliranje.* Zagreb: Školska knjiga.
10. Gallo, G., 2006. *Note di Simulazione.* Pisa: Universita di Pisa.
11. Gross, D., Thompson, J. M. & Shortle, J. F., 2018. *Fundamentals of Queueing Theory.* Hoboken: Wiley.
12. Karian, Z. A. & Dudewicz, E. J., 1998. *Modern Statistical, Systems, and GPSS Simulation.* Boca Raton: CRC Press.
13. Kaurić, S. & Pogarčić, I., 2013. Simulacija prodaje pretplatničkih karata javnoga prijevoza. *Zbornik Veleučilišta u Rijeci, Vol. 1,* pp. 307-318.
14. Matplotlib, 2022. *Matplotlib — Visualization with Python.* [Mrežno] Available at: https://matplotlib.org [Pokušaj pristupa 21. kolovoza 2022.].

15. Narayan Bhat, U., 2015. *An Introduction to Queueing Theory.* 2nd Edition ur. Boston: Birkhäuser.
16. Pidd, M., 2004. *Computer Simulation in Management Science.* Chichester: Wiley.
17. Robinson, S., 2014. *The Practice of Model Development and Use.* 2nd Edition ur. London: Palgrave Macmillan.
18. Sokolowski, J. A. & Banks, C. M., 2009. *Principles of Modeling and Simulation: A Multidisciplinary Approach.* Hoboken: Wiley.
19. Sztrik, J., 2021. *Basic Queueing Theory.* Debrecen: University of Debrecen.
20. Vagner, V., 2006. Poslovne simulacije. *Playmath,* IV(12), pp. 14-17.