

SIMULACIJA SISTEMA OPSLUŽIVANJA

Simulacija se rangira veoma visoko među široko primenjivanim tehnikama Operacionih Istraživanja. Osim toga, pošto simulacija predstavlja fleksibilnu, moćnu i nadasve intuitivnu tehniku, njena popularnost kontinuirano raste.

Ova tehnika uključuje korišćenje računara za *imitiranje* (simulaciju) rada čitavog procesa ili sistema. Npr. simulacija se često koristi pri analizi rizika finansijskih procesa tako što se u više navrata simulira tok novčanih transakcija da bi se generisao profil mogućih posledica. Takođe simulacija se široko primenjuje pri analizi stohastičkih procesa (sistema). Za takve sisteme, računar na slučajan način generiše i registruje pojavljivanje različitih događaja koji opisuju rad sistema na isti način kao kad bi on radio u stvarnosti. Zahvaljujući brzini, računar može da simulira čak i godine rada sistema i samo nekoliko sekundi. Zapisivanje performansi simuliranih operacija sistema, za nekoliko alternativnih konfiguracija sistema ili operativnih procedura, omogućuje vrednovanje i poređenje alternativa pre izbora jedne najpovoljnije.

Neke od najčešćih primena simulacije

Simulacija je izuzetno raznovrsna tehnika. Ona se može primeniti (sa različitim stepenom poteškoća) u virtuelnom istraživanju bilo koje vrste stohastičkih sistema. Upravo ta raznovrsnost je od simulacije načinila najšire korišćenu tehniku operacionih istraživanja pri analizi (istraživanju) ove vrste sistema, a njena popularnost i dalje raste.

Zbog ogromne raznolikosti svojih aplikacija, nije moguće nabrojati sve posebne oblasti u kojima se koristi simulacija. Neke od najvažnijih oblasti primene simulacije su: *1

- Projektovanje i upravljanje sistemima opsluživanja,
- Upravljanje zalihama,
- Procena verovatnoće završetka projekta do krajnjeg roka,
- Projektovanje i upravljanje proizvodnim sistemima,
- Projektovanje i upravljanje distributivnim sistemima,
- Analiza finansijskog rizika,
- Aplikacije u zdravstvu,
- Aplikacije u ostalim oblastima pružanja usluga, itd.

Suština simulacije

Simulacija je dugo vremena važan alat većine projektanata. Npr. simulacija leta aviona u vazdušnom tunelu je standardna praksa, kada se projektuje novi avion. Teorijski, zakoni fizike se mogu koristiti za dobijanje istih informacija o tome kako će se performanse aviona menjati u zavisnosti od varijacije projektnih parametara, ali u praksi ovu vrstu analize bilo bi veoma komplikovano izvesti (možda i nemoguće). Sledeća alternativa bila bi da se naprave pravi avioni sa različitim projektantskim rešenjima, što bi bilo vrlo skupo, kao i vrlo nesigurno. Stoga, posle sprovedenih preliminarnih teorijskih analiza da bi se dobio osnovni oblik (dizajn, nacrt) aviona, simulacija leta u vazdušnom tunelu je neophodan alat za eksperimentisanje sa različitim projektantskim rešenjima. Ova vrsta simulacije teži da imitira performanse pravog aviona u kontrolisanoj okolini u cilju da se proceni koje će biti njegove stvarne performanse. Pošto se, na ovaj način, razvije detaljan projekat aviona, prototip modela aviona sada može da se napravi i testira u stvarnim uslovima leta radi fine korekcije konačnog dizajna aviona.

Uloga simulacije u Operacionim Istraživanjima

Simulacija ima veoma važnu ulogu u mnogim oblastima operacionih istraživanja. Umesto da projektuju avion, tim (operacionih) istraživača radi na razvoju projekta ili operativne procedure za neki stohastički sistem (sistem koji se u vremenu ponaša po određenim zakonima verovatnoće). Neki od tih stohastičkih sistema liče na sisteme opsluživanja zasnovane na slučajnim procesima Markov-a dok su drugi daleko komplikovaniji. Umesto da koriste vazdušni tunel, rad stvarnog sistema se *imitira* koristeći raspodele verovatnoća da bi se na *slučajan način generisali* različiti događaji koji se javljaju u sistemu. Prema tome, simulacioni model sintetiše sistem, gradeći ga komponentu po komponentu i događaj po događaj. Na taj način model simulira rad sistema, u cilju dobijanja statističkih zakonitosti ponašanja sistema koje su rezultat različitih događaja generisanih na slučajan način. Pošto simuliranje rada sistema zahteva generisanje i obradu ogromne količine podataka, simulacioni eksperimenti se po pravilu izvode primenom računara.

Kada se simulacija koristi pri projektovanju stohastičkih sistema (sistema opsluživanja), obično joj prethode i prate je praktično isti koraci koji su opisani u slučaju projektovanja aviona. U ovom slučaju se takođe prvo urade preliminarne analize (najčešće sa najpribližnijim matematičkim modelom) da bi se dobila gruba konfiguracija sistema (uključujući i operativne procedure). Nakon toga se simulacija koristi da bi se eksperimentisalo sa različitim projektnim rešenjima i da bi se procenilo ponašanje svakog od njih. Pošto se na ovaj način završi detaljno projektovanje i odabere najbolje projektno rešenje, sistem se fizički postavlja i testira u stvarnim uslovima radi finog podešavanja projektovane konfiguracije sistema.

Detaljni simulacioni model kompleksnog sistema (opsluživanja) treba da bude tako formulisan da opiše rad sistema i na koji način će sistem biti simuliran. Simulacioni model ima nekoliko osnovnih blokova koji ga sačinjavaju: *2

1. Definisanje *stanja sistema* (tj. broj klijenata – jedinica u sistemu opsluživanja).
2. Identifikacija *moгуćih stanja* sistema.
3. Identifikacija *moгуćih događaja* (tj. dolazak jedinica u sistem i završetak opsluživanja) koji menjaju stanje sistema.
4. Definisanje *simulacionog sata*, lociranog na nekoj adresi u simulacionom programu, koji beleži prolazak (simulacionog) vremena.
5. Definisanje metoda za *slučajno generisanje događaja* raznih vrsta.
6. Formula za identifikaciju *promene stanja sistema* koja se generišu od strane raznih vrsta događaja.

Veliki napredak je napravljen u razvoju specijalnih softvera za efikasno integrisanje simulacionog modela u računarski program i nakon toga izvršavanje simulacije. Ipak, ako se radi o složenim (kompleksnim) sistemima, simulacija teži da bude relativno skupa procedura. Pošto se formuliše detaljni simulacioni model, prilično dugo vreme obično je potrebno da se razvije i testira računarski program potreban da bi se simulacija izvršila. Sledeće, mnogo sati rada računara, pri izvršavanju razvijenih simulacionih programa, je najčešće potrebno da bi se dobila dobra procena ponašanja svih alternativnih rešenja sistema. Konačno, svi dobijeni podaci treba da se detaljno analiziraju pre donošenja bilo kakvog konačnog zaključka. Ceo opisani proces oduzima mnogo vremena i truda. Prema tome simulaciju ne treba koristiti u slučaju kada manje skupe procedure mogu da obezbede iste informacije. *3

Simulacija se po pravilu koristi kada je razmatrani stohastički sistem previše komplikovan da bi se na zadovoljavajući način analizirao primenom analitičkih matematičkih modela (tj. modela teorije redova). Jedna od glavnih prednosti matematičkog modela je ta da on izdvaja suštinu problema i otkriva njegovu osnovnu strukturu, obezbeđujući na taj način uvid u povezanost uzroka i posledica unutar sistema. Prema tome, ako je projektant (kreator modela) u mogućnosti da napravi matematički model koji u sebi sadrži dopustiv nivo idealizacije problema i ako su dobijena rešenja prihvatljiva, ovaj pristup ima prvenstvo u odnosu na simulaciju. Ipak, veliki broj problema je previše komplikovan da bi se mogao primeniti gore opisani pristup. Tako da simulacija u najvećem broju slučajeva ostaje jedini praktični pristup rešavanju problema.

Diskretna ili Kontinualna Simulacija

Dve široke kategorije simulacije su tzv. diskretna i kontinualna simulacija.

Kontinualna simulacija je ona kod koje se stanja sistema kontinualno menjaju u vremenu. Npr. ako je sistem koji se razmatra avion u letu i ako se njegovo stanje definiše kao trenutna pozicija aviona u vazduhu, tada se stanje sistema (aviona) kontinualno menja u vremenu. Kontinualna simulacija se najčešće koristi pri projektovanju mašinskih sistema. *4

Kontinualna simulacija po pravilu zahteva korišćenje diferencijalnih jednačina da bi se opisalo intenzitet promene stanja sistema, što dovodi do toga da analiza može da bude veoma komplikovana.

Aproksimacijom kontinualnih promena stanja sistema povremenim diskretnim promenama, obično je moguće koristiti diskretnu simulaciju da bi se aproksimiralo ponašanje kontinualnih sistema. Opisana aproksimacija može u velikoj meri da pojednostavi analizu.

Diskretna simulacija je ona gde promene stanja sistema događaju trenutno u slučajnim vremenskim trenucima kao rezultat realizacije diskretnih događaja. Npr. u sistemima opsluživanja gde su stanja sistema određena brojem jedinica u sistemu, diskretni događaji koji menjaju tekuće stanje sistema su dolazak jedinica u sistem i odlazak jedinica iz sistema po završenom opsluživanju. Najveći broj simulacija u praksi su diskretne simulacije. *5

Postoje dva osnovna metoda koja se koriste za evidentiranje proticanja simulacionog vremena i događaja u sistemu. To su metoda *fiksnog povećanja vremena* i metoda *prelaska sa događaja na događaj*.

Metoda fiksnog povećanja vremena *6

1. *Proticanje (povećanje) simulacionog vremena* u sistemu se realizuje dodavanjem, na tekuće simulaciono vreme, male *konstantne vrednosti vremena* (npr. minut, sekund, deseti deo sekunde itd.).
2. *Ažuriranje sistema*, određuje se da li su se događaji odigrali u datom trenutku simulacionog vremena (događaji se odigravaju trenutno) i koje je rezultujuće stanje sistema. Takođe se beleže željene informacije u vezi sa performansama tj. karakteristikama sistema.

Kao što je u prethodnom tekstu rečeno, dva događaja se mogu odigrati u svakom trenutku simulacionog vremena, i to dolazak jedne jedinice u sistem i završetak opsluživanja jedinice koja se trenutno opslužuje. Nakon odigravanja bilo kog od ovih događaja određuje se novi vremenski trenutak odigravanja datog događaja generisanjem slučajnih brojeva na osnovu raspodela verovatnoća vremena između dolazaka jedinica u sistem i vremena opsluživanja, respektivno (eksponencijalne raspodele sa parametrom λ odnosno μ). Ako posle završetka opsluživanja nema

više jedinica u sistemu, određivanje vremena sledećeg završetka opsluživanja se odlaže do sledećeg dolaska jedinica u sistem.

Metoda prelaska sa događaja na događaj *7

Metoda **prelaska sa događaja na događaj** se razlikuje od metode fikskog povećanja vremena u tome što se simulaciono vreme u sistemu povećava za *promenljivu* vrednost a ne za konstantnu malu vrednost vremena. Ova promenljiva vrednost je jednaka razlici vremena između događaja koji se upravo odigrao i prvog sledećeg događaja, bilo koje vrste, koji treba da se odigra, tj. simulaciono vreme (sat) „skače“ od događaja do događaja.

1. Simulaciono vreme u sistemu se povećava (protiče) skokovito tj. od tekućeg vremena (događaja) do vremena odigravanja sledećeg događaja bilo koje vrste (dolazak jedinica ili opsluživanje).
2. *Ažuriranje sistema* se vrši određivanjem novog stanja sistema koje je nastalo odigravanjem datog događaja. Takođe se beleže željene informacije u vezi sa performansama tj. karakteristikama sistema.

Pri ovom načinu generisanja vremena u sistemu, potrebno je pratiti vremena izvršenja dva sledeća događaja tj. sledećeg dolaska jedinice i sledećeg završetka opsluživanja (ako se neka jedinica trenutno opslužuje). Ova vremena se određuju generisanjem slučajnih brojeva na osnovu raspodela verovatnoća vremena između dolazaka jedinica u sistem i vremena opsluživanja, respektivno. Drugim rečima, svaki put kada se dogodi dolazak jedinice u sistem ili se završi opsluživanje, određuje se vreme do sledećeg događaja (dolazak ili završetak opsluživanja) koje se dodaje trenutnom simulacionom vremenu i pamti kao vreme odigravanja sledećeg datog događaja (dolazak ili završetak opsluživanja). Ako posle završetka opsluživanja nema više jedinica u sistemu, određivanje vremena sledećeg završetka opsluživanja se odlaže do sledećeg dolaska jedinica u sistem. Da bi se odredilo koji će se događaj sledeći odigrati potrebno je naći minimum od zapamćenih vremena odigravanja sledećih događaja.

Najčešći početni uslovi pri izvođenju simulacije sistema opsluživanja su da je sistem u početnom trenutku prazan, što utiče na to da se vrednosti karakteristika sistema razlikuju od očekivanih vrednosti kada je sistem u stacionarnom režimu rada. Pošto je cilj simulacije sistema opsluživanja da se procene vrednosti karakteristika sistema opsluživanja u stacionarnom režimu rada, potrebno je neko vreme simulirati rad bez ažuriranja stanja sistema sve dok se praktično ne dostigne stacionarni režim rada sistema. Ovaj početni period čekanja da se praktično dostigne stacionarni režim rada sistema, pre početka ažuriranja stanja sistema, naziva se **period zagrevanja**. *8

U narednom tekstu biće reči samo o diskretnoj simulaciji.

Generisanje slučajnih brojeva

Implementacija simulacionog modela zahteva slučajne brojeve za generisanje slučajnih događaja koji su definisani odgovarajućim raspodelama verovatnoća. Jedna od metoda za generisanje slučajnih brojeva je korišćenje fizičkih uređaja kao što je npr. disk koji se okreće ili elektronskih uređaja. Nekoliko tabela slučajnih brojeva je generisano na ovaj način, uključujući i jednu koja sadrži 1 milion slučajnih brojeva, publikovana od strane Rand Corporation. Deo te tabele je prikazan u Tabeli VI-1. *9

Fizički uređaji su danas zamenjeni računarima kao osnovnim izvorima za generisanje slučajnih brojeva. Npr. *Microsoft Excel* koristi funkciju `RAND()` za tu svrhu. Mnogi drugi softverski paketi takođe imaju mogućnost da generišu slučajne brojeve potrebne u toku izvršavanja simulacije.

Tabela VI-1. Tabela slučajnih brojeva.

09656	96657	64842	49222	49506	10145	48455	23505	90430	04180
24712	55799	60857	73479	33581	17360	30406	05842	72044	90764
07202	96341	23699	76171	79126	04512	15426	15980	88898	06358
84575	46820	54083	43918	46989	05379	70682	43081	66171	38942
38144	87037	46626	70529	27918	34191	98668	33482	43998	75733
48048	56349	01986	29814	69800	91609	65374	22928	09704	59343
41936	58566	31276	19952	01352	18834	99596	09302	20087	19063
73391	94006	03822	81845	76158	41352	40596	14325	27020	17546
57580	08954	73554	28698	29022	11568	35668	59906	39557	27217
92646	41113	91411	56215	69302	86419	61224	41936	56939	27816
07118	12707	35622	81485	73354	49800	60805	05648	28898	60933
57842	57831	24130	75408	83784	64307	91620	40810	06539	70387
65078	44981	81009	33697	98324	46928	34198	96032	98426	77488
04294	96120	67629	55265	26248	40602	25566	12520	89785	93932
48381	06807	43775	09708	73199	53406	02910	83292	59249	18597
00459	62045	19249	67095	22752	24636	16965	91836	00582	46721
38824	81681	33323	64086	55970	04849	24819	20749	51711	86173
91465	22232	02907	01050	07121	53536	71070	26916	47620	01619
50874	00807	77751	73952	03073	69063	16894	85570	81746	07568
26644	75871	15618	50310	72610	66205	82640	86205	73453	90232

Karakteristike slučajnih brojeva

Procedura koju koriste računari za dobijanje slučajnih brojeva se naziva *generator slučajnih brojeva*.

Generator slučajnih brojeva je algoritam koji stvara nizove brojeva po zadatoj raspodeli verovatnoća na slučajan način. *10

U gornjoj definiciji *nizovi brojeva* znače da algoritam stvara mnogo slučajnih brojeva na serijski način. Premda pojedinačnom korisniku može da treba samo nekoliko brojeva, uopšteno algoritam mora da bude sposoban da stvara mnogo

slučajnih brojeva. *Raspodela verovatnoća* ukazuje na to da se odgovarajuća verovatnoća može dodeliti svakom slučajnom broju koji algoritam stvara.

Termin **slučajan broj** je rezervisan za realizaciju slučajne promenljive koja je definisana nekom formom *ravnomerne raspodele* verovatnoća, tako da su svi mogući brojevi *jednako verovatni*. Ako su za simulaciju potrebni slučajni brojevi koji odgovaraju nekim drugim zakonima raspodela, realizacija takve slučajne promenljive se naziva *slučajan broj generisan po datoj raspodeli*. *11

Slučajni brojevi se mogu podeliti u dve osnovne kategorije, slučajni celi brojevi i uniformni slučajni brojevi.

Slučajni celi brojevi predstavljaju realizaciju slučajne promenljive raspodeljene po *diskretnoj uniformnoj raspodeli* na intervalu $\underline{n}, \underline{n} + 1, \dots, \bar{n}$. U ovom slučaju zakon raspodele verovatnoća se definiše kao: *12

$$P(\underline{n}) = P(\underline{n} + 1) = \dots = P(\bar{n}) = \frac{1}{\bar{n} - \underline{n} + 1}.$$

Obično, $\underline{n} = 0$ ili 1 , što predstavlja podesne vrednosti za većinu aplikacija. (Ako \underline{n} ima neku drugu vrednost, tada se oduzimanjem \underline{n} ili $\underline{n} - 1$ od slučajnog celog broja donja granica intervala menja – svodi na 0 ili 1)

Uniformni slučajni brojevi predstavljaju realizaciju slučajne promenljive raspodeljene po (kontinualnoj) *ravnomernoj raspodeli* na intervalu $[a, b]$. Gustina ravnomerne raspodele definiše se kao: *13

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \wedge x > b \end{cases}$$

Ako vrednosti za a i b nisu specificirane, podrazumeva se da je $a = 0$ i $b = 1$.

Slučajni brojevi inicijalno generisani pomoću računara su obično slučajni celi brojevi. Međutim, ako je potrebno, ti brojevi se mogu jednostavno konvertovati u uniformne slučajne brojeve na sledeći način: *14

Ako se generisani slučajni ceo broj u intervalu od 0 do \bar{n} , podeli sa \bar{n} dobija se (približno) uniformni slučajan broj. (Ako je \bar{n} malo, ova aproksimacija treba da se poboljša dodavanjem $\frac{1}{2}$ slučajnom celom broju i tada se tako dobijen broj deli sa $\bar{n} + 1$ umesto sa \bar{n}).

Ovo predstavlja uobičajen metod za generisanje uniformnih slučajnih brojeva tj. slučajnih brojeva. Za ogromne vrednosti \bar{n} koje se obično koriste, ovo je u biti tačan metod.

Strogo govoreći, brojevi generisani od strane računara ne bi trebali da se zovu slučajni brojevi jer su oni u većoj ili manjoj meri predvidivi i ponovljivi (što je ponekad prednost), u zavisnosti od toga koji generator slučajnih brojeva se koristi. Otud, oni se ponekad nazivaju i **pseudo-slučajni brojevi**. Međutim, važno je da oni na zadovoljavajući način igraju ulogu slučajnih brojeva u simulacijama, ako je metod koji se koristi za njihovo generisanje valjan.

Različite relativno sofisticirane procedure postoje za testiranje da li generisani niz brojeva ima prihvatljivo pojavljivanje slučajnosti. U osnovi zahtevi su takvi da svaki sledeći broj u nizu ima jednaku verovatnoću uzimanja bilo koje od mogućih vrednosti i da bude statistički nezavisan od ostalih brojeva u nizu.

Kongruentni metod za generisanje slučajnih brojeva

Među velikim brojem generatora slučajnih brojeva koji su danas dostupni, najširu primenu imaju tzv. *kongruentni metodi* (aditivni, multiplikativni i kombinovani). Kombinovani kongruentni metod sadrži karakteristike ostala dva, tako da će u narednom tekstu prvo o njemu biti reči.

Kombinovani kongruentni metod generiše niz slučajnih celih brojeva u intervalu od 0 do $m - 1$. Metod uvek izračunava sledeći slučajni broj na osnovu poslednjeg dobijenog. Inicijalni slučajan broj x_0 , koji se naziva seme, može se odrediti koristeći neki publikovani izvor kao što je npr. Rand tabela (tabela VI-1). U praksi, kongruentni metodi izračunavaju $(n + 1)$ -vi slučajni broj x_{n+1} na osnovu n -tog slučajnog broja x_n koristeći sledeću rekurentnu formulu: *15

$$x_{n+1} \equiv (a \cdot x_n + c) \pmod{m}$$

gde a , c , i m su pozitivni celi brojevi ($a < m$, $c < m$). Ovaj matematički zapis ukazuje da x_{n+1} predstavlja *ostatak* deljenja $(a \cdot x_n + c)$ sa m . Prema tome, moguće vrednosti za x_{n+1} su 0, 1, ..., $m - 1$, što znači da m predstavlja željeni broj različitih vrednosti koje generator može da generiše kao slučajne brojeve.

Primeru radi, neka parametri a , c , i m imaju sledeće vrednosti $m = 8$, $a = 5$, $c = 7$, i neka je $x_0 = 4$. Rezultujući niz slučajnih brojeva je sračunat i prikazan u tabeli VI-2. (Niz ne može da se nastavi dalje zato što će brojevi početi da se ponavljaju istim redosledom).

Potrebno je naglasiti da ovaj niz sadrži svaku od osam mogućih brojeva tačno jedan put. Ova osobina je neophodna za generisanje niza slučajnih celih brojeva, ali se ona ne javlja pri svakoj kombinaciji parametara a i c (npr. $a = 4$, $c = 7$, i $x_0 = 3$). Srećna okolnost je da postoje pravila za izbor parametara a i c koja garantuju

ostvarenje ove osobine. (Ograničenja za seme x_0 ne postoje jer ono utiče samo na to koji će biti prvi broj u nizu).

Tabela VI-2. Prikaz kombinovanog kongruentnog metoda.

n	x_n	$5x_n + 7$	$(5x_n + 7)/8$	x_{n+1}
0	4	27	$3 + \frac{3}{8}$	3
1	3	22	$2 + \frac{6}{8}$	6
2	6	37	$4 + \frac{5}{8}$	5
3	5	32	$4 + \frac{0}{8}$	0
4	0	7	$0 + \frac{7}{8}$	7
5	7	42	$5 + \frac{2}{8}$	2
6	2	17	$2 + \frac{1}{8}$	1
7	1	12	$1 + \frac{4}{8}$	4

Broj uzastopnih slučajnih celih brojeva, pre nego što počnu da se ponavljaju istim redosledom, naziva se **dužina ciklusa**. U prethodnom primeru dužina ciklusa je osam. Maksimalna dužina ciklusa je m , tako da se razmatraju samo one vrednosti za parametre a i c koje obezbeđuju da se ostvari ciklus maksimalne dužine. *16

Tabela VI-3 prikazuje konverziju slučajnih celih brojeva u uniformne slučajne brojeve. Leva kolona sadrži slučajne cele brojeve dobijene u poslednjoj (desnoj) koloni tabele VI-2. Desna kolona tabele VI-3 daje odgovarajuće uniformne slučajne brojeve izračunate na osnovu formule:

$$\text{Uniformni slučajni broj} = \frac{1}{m} \cdot (\text{slučajni ceo broj} + \frac{1}{2}).$$

Potrebno je naglasiti da se uniformni slučajni brojevi nalaze na sredini jednog od osam jednakih intervala 0 do 0.125, 0.125 do 0.25, . . . , 0.875 do 1. Mala vrednost za m ($m=8$) ne dozvoljava da se dobiju druge vrednosti na intervalu $[0, 1]$, tako da je dobijena prilično gruba aproksimacija realnih uniformnih slučajnih brojeva. U praksi se generalno koriste mnogo veće vrednosti za m .

Tabela VI-3. Konverzija slučajnih celih brojeva u uniformne slučajne brojeve.

Slučajni celi brojevi	Uniformni slučajni brojevi
3	0.4375
6	0.8125
5	0.6875
0	0.0625
7	0.9375
2	0.3125
1	0.1875
4	0.5625

Za binarni računar sa veličinom *zapisa* od b bita, uobičajeni izbor za m je $m = 2^b$; što predstavlja ukupan broj nenegativnih celih brojeva koji se mogu izraziti u granicama kapaciteta veličine *zapisa*. (Bilo koji nepoželjni ceo broj koji se pojavi u nizu slučajnih brojeva se jednostavno ne uzima u obzir). Sa ovim izborom za m , izbor parametara a i c koji osigurava da će se svi mogući brojevi pojaviti tačno jednom pre nego što se bilo koji broj ponovi je sledeći: $a = 1, 5, 9, 13, \dots$ i $c = 1, 3, 5, 7, \dots$. Za decimalni računar sa veličinom *zapisa* od d cifara, uobičajeni izbor za m je $m = 10^d$, a ista osobina, kao i u prethodnom primeru, osigurava se izborom bilo koje od vrednosti $a = 1, 21, 41, 61, \dots$ i $c = 1, 3, 7, 9, 11, 13, 17, 19, \dots$ (svi pozitivni neparni brojevi *osim* onih čija je zadnja cifra 5).

Ponekad, je potrebno generisati slučajne cele brojeve sa relativno malim brojem cifara. Npr. pretpostavka je da su potrebno generisati slučajne brojeve sa samo tri cifre, tako da su moguće vrednosti sledeće: 000, 001, ... , 999. U takvom slučaju, uobičajena procedura je da se i dalje koriste vrednosti $m = 2^b$ ili $m = 10^d$, da bi se generisao veoma veliki niz slučajnih celih brojeva pre nego što bi slučajni celi brojevi počeli da se ponavljaju. U ovom slučaju generisani slučajni ceo broj, u izvornom obliku, se koristi samo u svrhu izračunavanja sledećeg slučajnog celog broja u datom nizu, jer mu se odbacuju sve cifre osim izabranih tri da bi se dobio traženi trocifreni slučajni ceo broj. Najčešće primenjivani princip je da se uzmu npr. poslednje tri cifre.

Multiplikativni kongruentni metod je samo specijalan slučaj kombinovanog kongruentnog metoda gde je $c = 0$. **Aditivni kongruentni metod** takođe je sličan, kod njega je $a = 1$ dok se c zamenjuje nekim slučajnim brojem koji u generisanom nizu prethodi x_n , npr, x_{n-1} (tako da je više od jednog semena potrebno da bi se pokrenulo izračunavanje slučajnih brojeva).

Među mogućim generatorima slučajnih brojeva baziranih na multiplikativnom kongruentnom metodu (izbor a i m), najšire je rasprostranjen tzv. *Learmouth-Lewis-ov* generator, koji generiše slučajne cele brojeve na sledeći način:

$$x_{n+1} \equiv 7^5 \cdot x_n \pmod{2^{31}-1}$$

Ovaj generator je testiran različitim statističkim testovima, i rezultati primenjenih statističkih testova pokazuju da su karakteristike generatora veoma zadovoljavajuće. Verzije ovog generatora se koriste u: IBM-ovoj verziji APL-a, paketu International Mathematics and Statistics Library (IMSL), paketu za generisanje slučajnih brojeva LLRANDOM itd. Tabele sa pogodnim vrednostima za semena su takođe dostupne.

Generisanje slučajnih brojeva prema zadatoj raspodeli verovatnoća

Postavlja se pitanje: kako na osnovu niza uniformnih slučajnih brojeva generisati niz slučajnih brojeva prema zadatoj raspodeli verovatnoća? Nekoliko različitih pristupa je poznato, u zavisnosti od prirode raspodele verovatnoća. Jedan od najčešće korišćenih metoda je tzv. *Metod inverzne transformacije*.

Metod inverzne transformacije *17

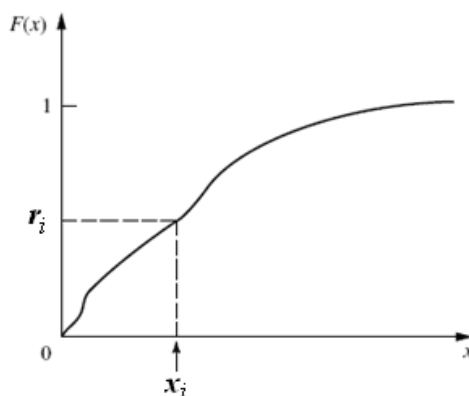
Metoda inverzne transformacije se može koristiti za generisanje slučajnih brojeva prema zadatoj raspodeli verovatnoća, bila ona diskretna ili kontinualna odnosno empirijska ili teorijska. Neka je X kontinualna slučajna promenljiva, tada se funkcija raspodele slučajne promenljive X definiše kao:

$$F(x) = P[X \leq x]$$

Generisanje slučajnih brojeva prema datoj raspodeli verovatnoća sastoji se iz dva koraka:

1. Generisanje uniformnog slučajnog broja r u intervalu između 0 i 1.
2. Postaviti da je $F(x) = r$ i rešiti po x , što tada predstavlja željeni slučajni broj generisan na osnovu zadate raspodele verovatnoća.

Ova procedura je prikazana na slici VI-1, gde se za datu realizaciju slučajne promenljive X (x_i), u zavisnosti od oblika funkcije raspodele $F(x)$, dobija se odgovarajući slučajni broj r_i .



Slika VI-1. Prikaz metoda inverzne transformacije.

Pri simulaciji sistema opsluživanja najčešće se koriste odgovarajuće empirijske raspodele (kontinualne i diskretne) i teorijske kontinualne raspodele (Ravnomerna, Eksponencijalna, Erlangova, Normalna raspodela).

Za određene kontinualne raspodele verovatnoća, metoda inverzne transformacije može se relativno jednostavno prilagoditi za računarsku primenu tako što se prvo jednačina $F(x) = r$ analitički reši po x .

Generisanje slučajnih brojeva, metodom inverzne transformacije, prema empirijskim kontinualnim i diskretnim raspodelama jednostavno se, primenom elementarnih logičkih operacija i linearnih interpolacija, prilagođava za računarsku primenu.

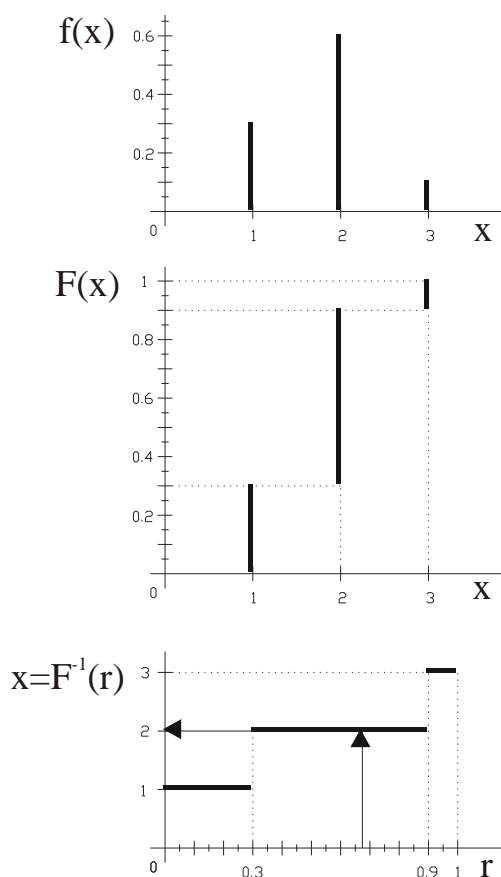
Generisanje slučajnih brojeva prema empirijskim raspodelama

Generisanje slučajnih brojeva po empirijskoj diskretnoj raspodeli *18

Neka je slučajna promenljiva X definisana diskretnim empirijskim zakonom raspodele verovatnoća:

$$X: \begin{pmatrix} 1 & 2 & 3 \\ 0,3 & 0,6 & 0,1 \end{pmatrix}. \quad (1)$$

Procedura generisanja slučajnih brojeva na osnovu diskretnog empirijskog zakona raspodele verovatnoća (1) je prikazana na slici VI-2, gde prvi dijagram prikazuje empirijski zakon raspodele – $f(x)$, drugi dijagram predstavlja kumulativnu funkciju empirijske raspodele – $F(x)$ i treći dijagram predstavlja inverznu kumulativnu funkciju empirijske raspodele – $x=F^{-1}(r)$ na osnovu koje se direktno određuju traženi slučajni brojevi, dok je r uniformni slučajan broj.

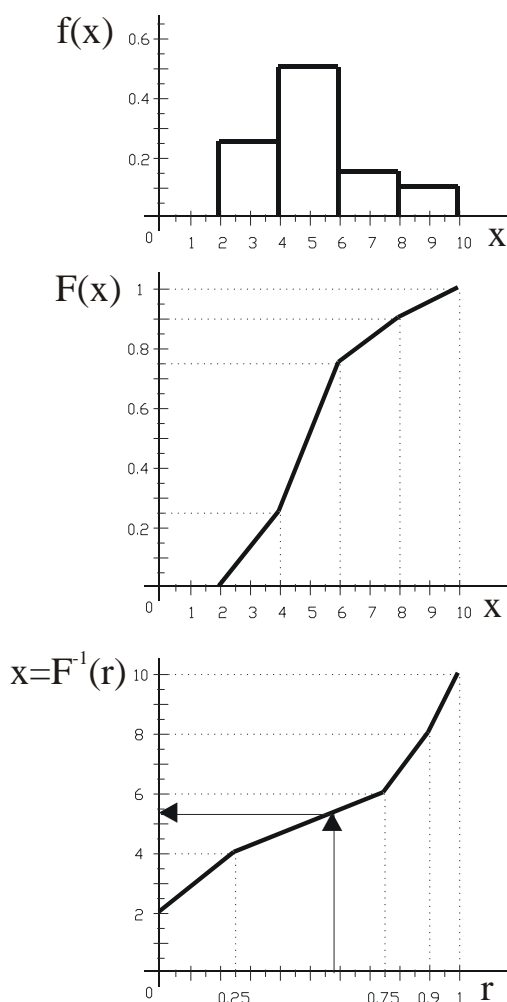


Slika VI-2. Generisanje slučajnih brojeva prema diskretnoj empirijskoj raspodeli.

Neka je slučajna promenljiva X definisana kontinualnom empirijskom raspodelom verovatnoća:

$$X: \begin{pmatrix} 2-4 & 4-6 & 6-8 & 8-10 \\ 0,25 & 0,5 & 0,15 & 0,1 \end{pmatrix}. \quad (2)$$

Procedura generisanja slučajnih brojeva na osnovu kontinualne empirijske raspodele verovatnoća (2) je prikazana na slici VI-3, gde prvi dijagram prikazuje histogram kontinualne empirijske raspodele – $f(x)$, drugi dijagram predstavlja kumulativnu funkciju empirijske raspodele – $F(x)$ i treći dijagram predstavlja inverznu kumulativnu funkciju empirijske raspodele – $x=F^{-1}(r)$ na osnovu koje se direktno određuju traženi slučajni brojevi, dok je r uniformni slučajan broj.



Slika VI-3. Generisanje slučajnih brojeva prema kontinualnoj empirijskoj raspodeli.



GenerisanjeSlučajnihBrojeva.Ink

Generisanje slučajnih brojeva prema teorijskim kontinualnim raspodelama

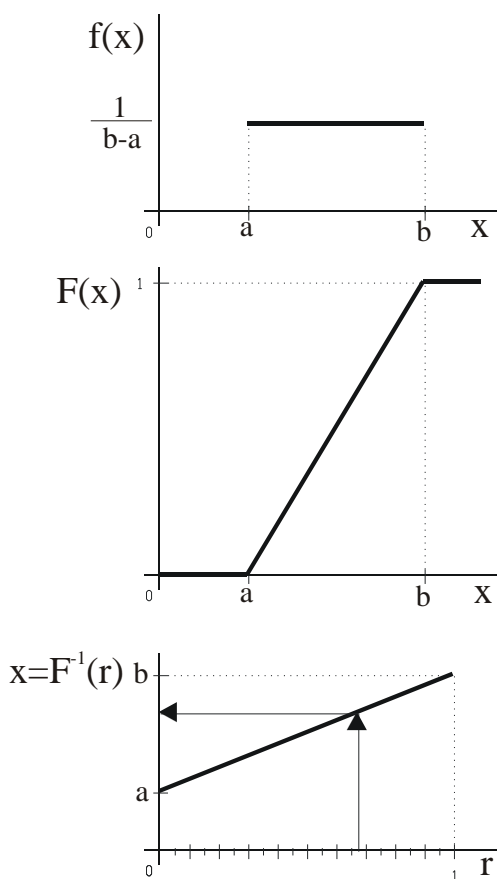
Generisanje slučajnih brojeva po ravnomernoj raspodeli *20

Gustina ravnomerne raspodele definiše se kao: (slika VI-4)

$$f(x) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \wedge x > b \end{cases} \quad (3)$$

dok je funkcija raspodele:

$$F(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x > b \end{cases} \quad (4)$$



Slika VI-4. Generisanje slučajnih brojeva po ravnomernoj raspodeli.

Matematičko očekivanje i disperzija ravnomerne raspodele su:

$$m = \frac{a+b}{2}, \quad \sigma^2 = \frac{(b-a)^2}{12}. \quad (5)$$

Inverzna funkcija raspodele, odnosno izraz na osnovu koga se dobijaju slučajni brojevi generisani po ravnomernoj raspodeli je sledeći:

$$F(x) = r = \frac{x-a}{b-a} \rightarrow F^{-1}(r) = a + (b-a) \cdot r \quad (6)$$

Na slici VI-4 prikazani su gustina raspodele $f(x)$, funkcija raspodele $F(x)$ i inverzna funkcija raspodele $F^{-1}(r)$ ravnomerne raspodele, gde je r uniformni slučajan broj.

Generisanje slučajnih brojeva po eksponencijalnoj raspodeli *21

Gustina eksponencijalne raspodele definiše se kao:

$$f(x) = \lambda \cdot e^{-\lambda \cdot x} \quad (7)$$

dok je funkcija raspodele:

$$F(x) = 1 - e^{-\lambda \cdot x} \quad (8)$$

Matematičko očekivanje i disperzija eksponencijalne raspodele su:

$$M[x] = m = \frac{1}{\lambda}, \quad D[x] = \sigma^2 = \frac{1}{\lambda^2}. \quad (9)$$

Inverzna funkcija raspodele, odnosno izraz na osnovu koga se dobijaju slučajni brojevi generisani po eksponencijalnoj raspodeli je sledeći:

$$F(x) = r = 1 - e^{-\lambda \cdot x} \rightarrow \ln(1-r) = -\lambda \cdot x \rightarrow x = -\frac{1}{\lambda} \cdot \ln(1-r)$$

Pošto se uniformni slučajan broj r generiše po ravnomernoj raspodeli u intervalu od 0 do 1, to je i slučajna promenljiva $1-r$ takođe raspodeljena po ravnomernoj raspodeli. Konačan izraz za slučajne brojeve generisane po eksponencijalnoj raspodeli je:

$$x = -\frac{1}{\lambda} \cdot \ln(r) \quad (10)$$

Generisanje slučajnih brojeva po Erlang-ovoj raspodeli *22

Neka su slučajne promenljive x_1 i x_2 raspodeljene po eksponencijalnoj raspodeli sa parametrom λ i gustinama raspodela:

$$f(x_1) = \lambda \cdot e^{-\lambda \cdot x_1}, \quad x_1 \geq 0, \text{ i}$$

$$f(x_2) = \lambda \cdot e^{-\lambda \cdot x_2}, \quad x_2 \geq 0.$$

Gustina raspodele nove slučajne promenljive z koja predstavlja zbir slučajnih promenljivih x_1 i x_2 dobija se na sledeći način:

$$z = x_1 + x_2 \rightarrow x_2 = z - x_1$$

$$f(z) = \int_0^z \lambda \cdot e^{-\lambda \cdot (z-x_1)} \cdot \lambda \cdot e^{-\lambda \cdot x_1} \cdot dx_1$$

gde prethodni integral predstavlja konvoluciju funkcija $f(x_1)$ i $f(x_2)$. Rešavanjem gornjeg integrala dobija se:

$$\begin{aligned} f(z) &= \int_0^z \lambda^2 \cdot e^{[-\lambda \cdot z - \lambda \cdot x_1 + \lambda \cdot x_1]} \cdot dx_1 = \int_0^z \lambda^2 \cdot e^{-\lambda \cdot z} \cdot dx_1 \\ &= \lambda^2 \cdot e^{-\lambda \cdot z} \cdot x_1 \Big|_0^z = \lambda^2 \cdot e^{-\lambda \cdot z} \cdot z - 0 \\ f(z) &= \lambda^2 \cdot z \cdot e^{-\lambda \cdot z} \end{aligned} \quad (11)$$

Izraz (11) predstavlja gustinu Erlang-ove raspodele drugog stepena sa parametrom λ . Uopšteno Erlang-ova raspodela stepena k sa parametrom λ se može dobiti sabiranjem k slučajnih promenljivih raspodeljenih po eksponencijalnoj raspodeli sa istim parametrom λ . Ova činjenica se koristi za generisanje slučajnih brojeva raspodeljenih po Erlang-ovoj raspodeli.

Gustina Erlang-ove raspodele stepena k definiše se kao:

$$f(x) = \frac{1}{(k-1)!} \cdot \lambda^k \cdot x^{k-1} \cdot e^{-\lambda \cdot x} \quad (12)$$

dok je funkcija raspodele:

$$F(x) = 1 - \sum_{i=0}^{k-1} \frac{(\lambda \cdot x)^i}{i!} \cdot e^{-\lambda \cdot x} \quad (13)$$

Matematičko očekivanje i disperzija eksponencijalne raspodele su:

$$M[x] = m = \frac{k}{\lambda}, \quad D[x] = \sigma^2 = \frac{k}{\lambda^2}. \quad (14)$$

Slučajan broj z generisan po Erlang-ovoj raspodeli stepena k i parametrom λ može se dobiti sabiranjem k slučajnih promenljivih x_1, x_2, \dots, x_k raspodeljenih po eksponencijalnoj raspodeli sa parametrom λ , kao:

$$\begin{aligned} z = x_1 + x_2 + \dots + x_k &= -\frac{1}{\lambda} \cdot \ln(r_1) - \frac{1}{\lambda} \cdot \ln(r_2) - \dots - \frac{1}{\lambda} \cdot \ln(r_k) \\ &= -\frac{1}{\lambda} \cdot [\ln(r_1) + \ln(r_2) + \dots + \ln(r_k)] \\ z &= -\frac{1}{\lambda} \cdot \ln(r_1 \cdot r_2 \cdot \dots \cdot r_k) \end{aligned} \quad (15)$$

što znači da je za generisanje jednog slučajnog broja po Erlang-ovoj raspodeli stepena k , potrebno k uniformnih slučajnih brojeva.

Generisanje slučajnih brojeva po Normalnoj raspodeli *23

Gustina Normalne raspodele definiše se kao:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{1}{2 \cdot \sigma^2} \cdot (x-m)^2} \quad (16)$$

gde je m – matematičko očekivanje a σ srednje kvadratno odstupanje.

Svaka Normalna raspodela sa parametrima m i σ , $N(m, \sigma)$, može se svesti na tzv. standardizovanu Normalnu raspodelu sa parametrima 0 i 1, $N(0,1)$, na sledeći način:

$$x(0,1) = \frac{x(m, \sigma) - m}{\sigma}.$$

Gustina standardizovane Normalne raspodele definiše se kao:

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{x^2}{2}}$$

Generisanje slučajnih brojeva po Normalnoj raspodeli nije moguće izvršiti direktno primenom metode inverzne transformacije, jer funkciju raspodele Normalne raspodele nije moguće odrediti u analitičkom obliku, već je potrebno primeniti *centralnu graničnu teoremu* koja u ovom slučaju može da se formuliše kao:

Za slučajni uzorak r_1, r_2, \dots, r_n izabran iz skupa čija raspodela ima matematičko očekivanje m i disperziju σ^2 , raspodela srednje vrednosti uzorka \bar{r}_n je slučajna promenljiva čija raspodela teži Normalnoj raspodeli sa matematičkim očekivanjem m i disperzijom σ^2/n , kad $n \rightarrow \infty$.

Svođenje Normalne raspodele sa matematičkim očekivanjem m i disperzijom σ^2/n na standardizovanu Normalnu raspodelu $N(0,1)$ može se izvršiti na sledeći način:

$$x(0,1) = \frac{\bar{r}_n - m}{\frac{\sigma}{\sqrt{n}}} = \frac{\frac{1}{n} \cdot \sum_{i=1}^n r_i - m}{\frac{\sigma}{\sqrt{n}}} = \frac{\sum_{i=1}^n r_i - n \cdot m}{\sigma \cdot \sqrt{n}}.$$

Pošto se uniformni slučajni brojevi generišu po Ravnomernoj raspodeli na intervalu od 0 do 1 to matematičko očekivanje i disperzija imaju vrednosti: (izraz (5)):

$$m = \frac{1}{2}, \quad \sigma^2 = \frac{1}{12}.$$

Zamenom prethodnih vrednosti u izraz za $x(0,1)$ dobija se:

$$x(0,1) = \frac{\sum_{i=1}^n r_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}}$$

odakle se za $n=12$ dobija konačan izraz za generisanje slučajnih brojeva po standardizovanoj Normalnoj raspodeli $N(0,1)$ kao:

$$x(0,1) = \sum_{i=1}^{12} r_i - 6,$$

što znači da je za generisanje jednog slučajnog broja po standardizovanoj Normalnoj raspodeli $N(0,1)$ potrebno dvanaest uniformnih slučajnih brojeva.

Slučajan broj generisan po Normalnoj raspodeli sa parametrima m i σ , $N(m,\sigma)$ može se dobiti kao:

$$x(m,\sigma) = \sigma \cdot x(0,1) + m.$$

Pregled osnovnih simulacionih koraka

U narednom tekstu biće dat pregled osnovnih koraka uključenih u najveći broj oblasti operacionih istraživanja baziranih na primeni simulacije. *24

Korak 1: Formulisanje problema i planiranje istraživanja

Istraživački tim treba da započne svoj rad sastankom sa menadžmentom da bi dobio odgovore na sledeća pitanja:

1. Koji problem menadžment hoće da reši ?
2. Šta su krajnji ciljevi istraživanja ?
3. Na koje probleme posebno treba obratiti pažnju ?
4. Koje alternativne konfiguracije sistema treba razmatrati ?
5. Koji pokazatelji performansi sistema su od interesa za menadžment ?
6. Koja su vremenska ograničenja za izvođenje simulacije ?

Pored sastanka sa menadžmentom, tim takođe treba da se sastane sa inženjerima i operativnim osobljem kako bi saznao detalje vezane za rad sistema. (Tim takođe treba da uključi u rad jednog ili više zaposlenih koji iz prve ruke imaju znanje o radu sistema – izvršioce). Ako je trenutna konfiguracija sistema operativna tj. ako sistem radi, tim treba da snima rad sistema da bi se identifikovale komponente sistema i njihova međusobna povezanost.

Pre nego što se zaključe aktivnosti vezane za ovaj prvi korak, rukovodilac istraživačkog tima treba da isplanira kompletno istraživanje u smislu: broja ljudi, njihovih odgovornosti, rasporeda aktivnosti i budžeta istraživanja.

Korak 2: Prikupljanje podataka i formulisanje simulacionog modela

Tip potrebnih podataka zavisi od prirode sistema koji treba da se simulira. Za sistem opsluživanja, potrebni podaci su raspodela vremena dolaska i raspodela vremena opsluživanja. Za sistem upravljanja zalihama, potrebni podaci su raspodela zahteva za datim proizvodom kao i raspodela vremena između ispostavljanja zahteva za obnavljanjem zaliha i isporuke naručenih količina. Za upravljanje projektima po metodi PERT, gde je trajanje aktivnosti neodređeno, potrebne su raspodele vremena trajanja pojedinih aktivnosti. Za proizvodni sistem, koji se sastoji od mašina koje mogu da otkazu u vremenu, potrebni podaci su raspodele bezotkaznog rada pojedinih mašina kao i raspodele vremena potrebnog za otklanjanje kvarova.

U svakom od navedenih primera potrebni podaci su raspodele verovatnoća odgovarajućih slučajnih događaja. Da bi se generisao reprezentativni scenario kako će sistem funkcionisati, neophodno je da se simulacijom generišu relevantni slučajni događaji na osnovu usvojenih raspodela verovatnoća, radije nego da se za simulaciju rada sistema koriste srednje vrednosti trajanja pojedinih događaja.

Uopšteno, jedino je moguće proceniti odgovarajuće raspodele. To se može uraditi ako se posmatra (snima) rad postojeće konfiguracije sistema, koji se razmatra, ili nekog sličnog sistema. Nakon što se prikupi dovoljna količina (broj) podataka – uzorak, koriste se odgovarajući statistički testovi za verifikaciju pripadnosti uzorka datoj teorijskoj raspodeli verovatnoća (npr. χ^2 – test). Srednja vrednost i disperzija (srednje kvadratno odstupanje) uzorka se uzimaju za srednju vrednost i disperziju teorijske raspodele na koju se uzorak testira. Ako se odgovarajući podaci ne mogu dobiti zato što ne postoje slični sistemi, drugi mogući izvori informacija za procenu odgovarajuće raspodele mogu da budu: operativna uputstva, iskustveni inženjerski podaci, intervjui sa osobama koje imaju iskustva u radu sa sličnim vrstama operacija itd.

Simulacioni model se često prikazuje (formuliše) u obliku algoritma koji povezuje različite komponente sistema u celinu. Operativna pravila se definišu za svaku

komponentu sistema, uključujući i raspodele verovatnoća koje određuju kada će se odgovarajući događaji odigrati. Model treba da sadrži dovoljno detalja da zadovoljavajući način opiše suštinu rada sistema. Za analize – simulacije velikih sistema, pokazalo se kao korisno da se krene sa formulisanjem i testiranjem jednostavnijih verzija modela pre nego što se u njega dodaju važni detalji.

Korak 3: Provera ispravnosti simulacionog modela

Pre početka rada na pisanju računarskog programa, istraživački tim treba da angažuje ljude koji su do tančina upoznati kako bi sistem trebalo da radi da bi se proverila ispravnost (tačnost) simulacionog modela. U najvećem broju slučajeva to se radi prikazivanjem odnosno analizom koncepta modela, pred svim bitnim činionicima (ljudima) vezanim za projekat. Na tipičnim takvim sastancima obično, nekoliko pogrešnih pretpostavki u modelu će biti otkriveno i korigovano, nekoliko novih pretpostavki će biti dodato, i neki broj problema (pitanja) će biti rešeno na drugačiji način u smislu koliko je potrebno detaljno modelirati različite delove sistema.

Korak 4: Izbor softvera i pisanje računarskog programa

Četiri osnovne klase softvera se koriste za računarske simulacije. Jedna od najjednostavnijih su tzv. tabelarni softveri (*spreadsheet software*), čiji je najrasprostranjeniji predstavnik *Microsoft Excel*. *25

Ostale tri klase simulacionih softvera su namenjene za ozbiljnije primene gde nije više podesno koristiti tabelarne softvere. Jedna takva klasa su *programski jezici opšte namene*, kao što su C++, FORTRAN, PASCAL, BASIC itd. Ovi programski jezici (i njihove procedure) su korišćeni u počecima razvoja simulacije kao discipline, zbog njihove velike fleksibilnosti i mogućnosti programiranja bilo koje vrste simulacije. Međutim, zbog značajnog vremena potrebnog za programiranje, oni (programski jezici) se ne koriste ni približno kao pre.

Treća klasa su tzv. simulacioni jezici opšte namene. Ovi simulacioni jezici pružaju mnogo pogodnosti pri pisanju simulacionih programa, i na taj način značajno smanjuju vreme potrebno za programiranje. Oni takođe predstavljaju pogodno okruženje za simulaciono modeliranje. Iako manje fleksibilni od programskih jezika opšte namene, njihovim korišćenjem se može napisati program za skoro svaku vrstu simulacionog modela. Ipak, određeni stepen iskustva, u radu sa simulacionim jezikom, je potreban.

Najpoznatiji simulacioni jezici opšte namene su GPSS, SIMSCRIPT, SLAM, SIMAN itd. Inicijalne verzije ovih simulacionih jezika datiraju još iz 1961, 1963, 1979, i 1983 godine respektivno, ali su svi oni prošli test vremena.

Ključni razvoj u oblasti simulacije u 80-tim i 90-tim godinama prošlog veka je pojava četvrte klase softvera, nazvane *aplikativno-orijentisani simulatori* ili kraće samo **simulatori**. Svaki od simulatora je dizajniran – namenjen za simuliranje specifične vrste sistema, kao što su: određene vrste proizvodnog procesa, računari, komunikacioni sistemi itd. Neki od njih su veoma specifični – usko orijentisani kao npr. za simulaciju: naftnih i gasnih bušotina, rada nuklearne elektrane ili kardio vaskularne fiziologije itd. Njihov cilj je da obezbede mogućnosti da se načini simulacioni program koristeći samo menije, tabele i grafike tj. bez potrebe za klasičnim programiranjem. Njihova primena se relativno lako savladava i imaju okruženje, potrebno za modeliranje, veoma blisko predmetnom sistemu.

Simulator može biti veoma moćno sredstvo, ako sistem koji treba da se simulira pripada propisanoj kategoriji sistema za koju je simulator dizajniran. Ipak, spisak dostupnih karakteristika sistema, koje se mogu pratiti, je veoma ograničen. Zbog toga, osnovni nedostatak većine simulatora je da su oni ograničeni samo na modeliranje onih konfiguracija sistema, koje njihove standardne mogućnosti dozvoljavaju. Neki simulatori dozvoljavaju mogućnost ugrađivanja rutina (procedura) u simulacioni model, pisanih u nekom od programskih jezika opšte namene, radi modeliranja nestandardnih događaja, situacija, veza itd. Ova opcija je često potrebna kada se simulira rad relativno složenih sistema.

Sledeći ključni napredak u poslednjih nekoliko godina je bio razvoj **animacije** za prikazivanje računarske simulacije u akciji. U animaciji računarske simulacije, osnovni elementi sistema se prikazuju na ekranu (računara) pomoću ikona koje menjaju oblik, boju, ili poziciju kada se menja stanje simuliranog sistema. Najveći broj prodavaca simulacionih softvera danas nude verzije njihovih softvera sa mogućnostima animacije. Štaviše, animacija je postala do te mere detaljna, da pruža u nekim slučajevima čak i trodimenzionalni prikaz simuliranog procesa.

Osnovni razlog popularnosti animacije je u njenoj mogućnosti da saopšti suštinu simulacionog modela (ili datog simulacionog eksperimenta) menadžerima i ostalim ključnim ljudima, na jednostavan način, što veoma povećava kredibilitet simulacionog pristupa. Animacija takođe može biti korisna u otklanjanju grešaka u računarskom programu simulacionog modela.

Korak 5: Testiranje valjanosti simulacionog modela

Nakon što se računarski program napiše i pošto se otklone greške, sledeći važan korak je testiranje da li simulacioni model ugrađen u program daje valjane (dobre)

rezultate za sistem koji reprezentuje. Posebno, da li će pokazatelji performansi realnog sistema biti dovoljno dobro aproksimirani vrednostima, tih istih pokazatelja performansi, generisanim od strane simulacionog modela ?

Na ovo pitanje je obično teško dati odgovor jer većina verzija „realnog“ sistema trenutno ne postoji, što dovodi to toga da je tad svrha simulacije da istraži i uporedi različite predložene konfiguracije sistema i pomogne u izboru najbolje.

Međutim, neke verzije realnog sistema mogu trenutno da budu u radu. Ako je to slučaj, podaci o performansama realnog sistema treba da se uporede sa odgovarajućim izlaznim veličinama generisanih od simulacionog programa.

U nekim slučajevima, matematičkim modelom se mogu opisati neke jednostavnije verzije sistema. Tada, rezultati dobijeni primenom matematičkog modela takođe se trebaju uporediti sa rezultatima simulacije.

Kada ne postoje realni podaci koji bi se poredili sa rezultatima simulacije, jedna od mogućnosti je sprovođenje *ogleda* u cilju prikupljanja potrebnih podataka. Sprovođenje ogleda podrazumeva konstruisanje malog prototipa neke od verzija razmatranog sistema i njegovo puštanje u rad. Takav prototip se, posle završene simulacije, takođe može koristiti i za fino podešavanje sistema pre puštanja stvarnog (realnog) sistema u rad.

Još jedan koristan test valjanosti jeste nepristrasna principska provera, kako se rezultati simulacije menjaju u zavisnosti od promene konfiguracije simuliranog sistema, od strane stručnog operativnog osoblja. Čak i kada ne postoje osnove za proveru zakonitosti promene pojedinih pokazatelja performansi dobijenih za datu verziju sistema, neki zaključci se mogu izvući, kao što je npr. relativna promena performansi sistema u zavisnosti od promene konfiguracije (parametara) sistema.

Gledanje animacije računarske simulacije je takođe način da se ispita valjanost simulacionog modela. Od trenutka kada model počne ispravno da radi, animacija takođe stvara zainteresovanost i poverenje za simulaciono istraživanje kako kod menadžmenta tako i kod operativnog osoblja.

Korak 6: Planiranje izvođenja simulacije

Na ovom mestu se donosi odluka koje konfiguracije sistema će biti simulirane. To je često interaktivni proces gde se početni rezultati, za gamu razmatranih konfiguracija sistema, koriste za donošenje odluke o tome koje konfiguracije sistema opravdavaju dalja detaljna ispitivanja.

Takođe u ovom trenutku potrebno je doneti i neke odluke vezane za statistička pitanja vezana za simulaciju. Jedno od takvih pitanja je i *dužina perioda zagrevanja*. Kao što je već rečeno, ovaj period u stvarnosti predstavlja vreme potrebno da sistem dostigne stacionarni režim rada. Tek po isteku perioda zagrevanja počinje se sa prikupljanjem podataka generisanih od strane simulacionog programa. Preliminarni simulacioni eksperimenti se često koriste da bi se ovaj problem analizirao. Pošto složeni sistemi često zahtevaju iznenađujuće dugo vreme za dostizanje stacionarnog režima rada, od pomoći je da se za početno stanje sistema koji se simulira, izabere takvo stanje koje je približno stanju sistema u stacionarnom režimu rada, da bi se, što je moguće više, skratilo vreme potrebno za period zagrevanja.

Još jedno važno statističko pitanje jeste i *dužina trajanja simulacije* koja sledi posle perioda zagrevanja za svaku konfiguraciju sistema koja se simulira. Treba imati na umu da simulacija ne daje *tačne* vrednosti pokazatelja performansi sistema. Umesto toga, svaka simulacija date konfiguracije sistema može se posmatrati kao *statistički eksperiment* koji generiše jednu *slučajnu (statističku) realizaciju* pokazatelja performansi simuliranog sistema. Ove realizacije (uzorak) se koriste za određivanje *statističke procene*, najčešće matematičkog očekivanja, pokazatelja performansi sistema. Produžavanjem vremena trajanja simulacije povećava se i preciznost ovih procena. *26

Broj simulacionih eksperimenata (n) tj. broj simulacija date konfiguracije sistema takođe veoma utiče na preciznost statističkih procena matematičkog očekivanja tj. srednje vrednosti pokazatelja performansi sistema. Standardno odstupanje srednje vrednosti uzorka, koga sačinjavaju realizacije pokazatelja performansi simuliranog sistema, obrnuto je proporcionalno kvadratnom korenu veličine uzorka tj. broju simulacionih eksperimenata $\sim \sigma / \sqrt{n}$ (gde je σ standardno odstupanje jednog simulacionog eksperimenta) *27. Prema tome, potrebno je veliko povećanje veličine uzorka, tj. broja simulacionih eksperimenata, da bi se ostvarilo malo povećanje u preciznosti procene matematičkog očekivanja pokazatelja performansi simuliranog sistema.

Statistička teorija potrebna za izvođenje statističkog eksperimenta vezanog za simulaciju se malo razlikuje od teorije potrebna za izvođenje statističkih eksperimenata vezanih direktno za posmatranje (pokazatelja) performansi fizičkog sistema. Zbog toga, uključenje profesionalnog statističara (ili u najmanju ruku iskusnog simulacionog analitičara sa velikim statističkim znanjem) u istraživački tim može biti dragoceno u ovom koraku.

Korak 7: Vođenje simulacionog eksperimenta i analiza rezultata

Izlazne veličine iz simulacionog eksperimenta daju statističke procene željenih (praćenih) pokazatelja performansi za svaku simuliranu konfiguraciju sistema. Kao dopuna *tačkaste procene* svakog pokazatelja, može se odrediti *interval poverenja* koji ukazuje na raspon verovatnih vrednosti pokazatelja performansi sistema. *28

Dobijeni rezultati mogu odmah da ukažu na onu konfiguraciju sistema koja je evidentno bolja od ostalih. Mnogo češće, dobijeni rezultati će identifikovati nekoliko ozbiljnih kandidata koji mogu da budu najbolji. Za bolje poređenje kandidata potrebno je izvesti dodatne simulacije koje duže traju. Dodatne simulacije se takođe mogu koristiti za fino podešavanje određenih detalja izabrane najbolje konfiguracije sistema.

Korak 8: Prezentacija i preporuka menadžmentu

Posle završavanja analize, istraživački tim treba da prezentuje rezultate istraživanja, zaključke do kojih je došao i da odgovarajuće preporuke menadžmentu. Ovo se obično radi i u formi pisanog izveštaja i u formi formalne prezentacije istraživanja menadžerima odgovornim za donošenje odluka vezanih za sistem koji se analizira.

Izveštaj i prezentacija trebaju da na jednostavan ali razumljiv način prikažu kako je analiza (eksperiment) vođena, uključujući tu i dokumenta koja potvrđuju valjanost simulacionog modela. Animirani prikaz rada računarske simulacije, ako je moguće, bi trebalo uključiti u prezentaciju, da bi se bolje prikazao simulacioni proces i dobio kredibilitet. Numerički rezultati koji daju obrazloženje za preporuke takođe moraju biti uključeni u prezentaciju i izveštaj.

Menadžment obično angažuje istraživački tim u početnim fazama implementacije novog sistema, uključujući tu i podučavanje angažovanog osoblja.

PITANJA:

1. Navesti neke od najvažnijih oblasti primene simulacije.
2. Osnovni blokovi simulacionih modela.
3. U kojim slučajevima treba primenjivati simulaciju.
4. Na koje sisteme se odnosi kontinualna simulacija.
5. Na koje sisteme se odnosi diskretna simulacija.
6. Metoda fiksnog povećanja vremena.
7. Metoda prelaska sa događaja na događaj.
8. Šta predstavlja period zagrevanja simulacionog modela.
9. Načini generisanja slučajnih brojeva.
10. Definicija generatora slučajnih brojeva.
11. Definirati termin slučajan broj.
12. Slučajni celi brojevi.
13. Uniformni slučajni brojevi.
14. Konverzija celih slučajnih brojeva u cele slučajne brojeve.
15. Kombinovani kongruentni metod za generisanje slučajnih celih brojeva.
16. Šta predstavlja dužina ciklusa generatora slučajnih celih brojeva.
17. Metod inverzne transformacije za generisanje slučajnih brojeva prema zadatoj raspodeli verovatnoća.
18. Generisanje slučajnih brojeva po empirijskoj diskretnoj raspodeli.
19. Generisanje slučajnih brojeva po empirijskoj kontinualnoj raspodeli.
20. Generisanje slučajnih brojeva po ravnomernoj raspodeli.
21. Generisanje slučajnih brojeva po eksponencijalnoj raspodeli.
22. Generisanje slučajnih brojeva po Erlang-ovoj raspodeli.
23. Generisanje slučajnih brojeva po Normalnoj raspodeli.
24. Osnovni simulacioni koraci.
25. Simulacioni softveri.
26. Od čega zavisi tačnost dobijenih rezultata simulacije.
27. Čemu je proporcionalna greška kod simulacionog eksperimenta.
28. Način prikazivanja pokazatelja performansi sistema, dobijenih simulacijom.