



UNIVERZITET  
**“DŽEMAL BIJEDIĆ”**  
U MOSTARU

# SIMULACIJA KAMIONA ZA DOSTAVU UGLJA

Predmet : Modeliranje & Simulacija

Ime student : Armin Smajlagić

Broj indeksa studenta : -//-

Smjer : Softverski Inženjering

Nastavni Profesor : -//-

## Sadržaj

1. Uvod .....	3
1.1 Postavka simulacije .....	3
1.2 Cilj simulacije (Mjere performansi) .....	4
2. Glavni dio rada, Opis metode .....	5
2.1 Prepoznavanje glavnih komponenti simulacije .....	5
2.2 Vršenje simulacije .....	6
2.3 Opis Metode .....	11
3. Opis rezultata, dijagrami .....	12
3.1 Estimacija iskorištenja vagi (Glavna mjera performansi) .....	12
3.2 Dijagrami .....	13
4. Budući rad .....	14
5. Zaključak .....	15
6. Literatura .....	16

---

## Abstrakt

*U ovom radu je modeliranje i simulacija poslužilo kao rješenje analize problema firme za dostavu uglja koja ne zna da li imaju dovoljno ili previše kamiona, vagi ili punjača uglja. Problem je postavljen na generički način kako bi se njegovo rješenje moglo iskoristiti i na druge probleme u oblasti sistema sa redovima (eng. Queueing systems). Podaci korišteni za vršenje simulacije nisu empirijski već su generisani matematičkim metodama prevedenim u programski kod jezika C++. Postignuti rezultati su estimacije zasnovane na jednom ciklusu simulacije pa je bitno reći da se radi o short-running simulaciji. Kvalitetnija obrazloženja spomenutih odluka kao trajanje simulacije, generisanje slučajnih i sl. se nalaze u odgovarajućim poglavljem ovog rada.*

---

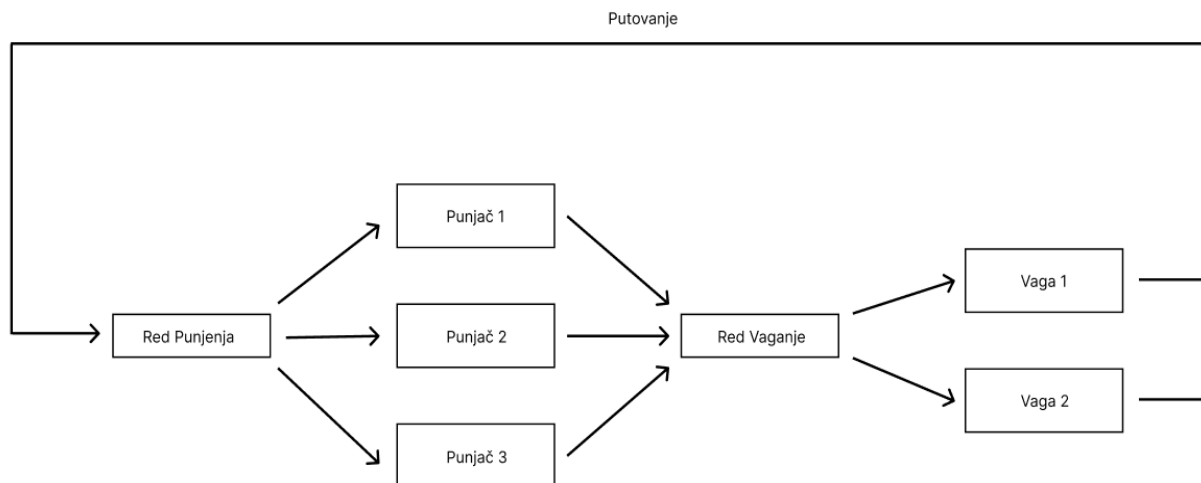
# 1. Uvod

## 1.1 Postavka simulacije

Ideja ove simulacije jeste predstaviti konačan broj kamiona koji bivaju punjeni i potom vagani na glavnoj stanici. Po završetku punjenja i vaganja šalju se na putovanje (u trajanju od proizvoljnog vremenskog perioda) kako bi se ispraznili i potom ponovno stižu na glavnu stanicu gdje se proces ponavlja (bivaju punjena pa vagani). Ovo je simulacija sa 2 reda za čekanje koja su povezane u seriju. Prvi red za čekanje će biti namijenjen za proces punjenja, a drugi red za vaganje. Oba reda funkcionišu po principu first-in-first-out ili first-come-first-served. Metod korišten za generisanje slučajnih za vrijeme trajanja punjenja, vaganja i putovanja će biti naknadno definisan.

Postavka iznad na generički način opisuje problem, pa ču u nastavku promijeniti inicijalne uslove kao broj kamiona, broj vagi i broj punjača dok ču same redove ostaviti iste kako bi zadržalo svojstvo single-queue simulacije (i ako imaju 2 reda radi se o single-queue jer su povezani u seriju a ne paralelno).

Moja simulacija će imati 8 kamiona koji bivaju punjeni na tri punjača i važu se na dvije vage kao na slici ispod (slika 1.)



Slika 1. Problem kamiona za dostavu uglja

## 1.2 Cilj simulacije (Mjere performansi)

Cilj je da se napravi estimacija iskorištenja vage i punjača (eng. loadera) za ukupno vrijeme trajanja simulacije. Broj vagi i punjača će uticati na krajnji rezultat estimacije njihove iskorištenosti. Spomenuta estimacija iskorištenja vagi i punjača se kroz terminologiju simulacije diskretnih događaja može se promatrati kao mjera performansi poznata kao iskorištenje servera (eng. server utilization).

Kako bi izvršili estimaciju iskorištenja vagi i punjača neophodno je pamtiti i evidentirati pri samoj simulaciji sljedeće kumulativne statistike :

- $B_P$  – ukupno vrijeme zauzetosti tri punjača (od početka simulacije do izlaska posljednjeg kamiona)
- $B_V$  – ukupno vrijeme zauzetosti dvije vage (od početka simulacije do izlaska posljednjeg kamiona)

Spomenute statistike ovise o odabiru distribucija i metode generisanja slučajnih za vrijeme vaganja i vrijeme punjenja.

Dodatno bitne vrijednosti za računanje iskorištenosti su :

- $T$  – ukupno vrijeme trajanja simulacije
- $N_P$  – broj punjača
- $N_V$  – broj vagi

Formule po kojima ću računati spomenute statistike glase :

- Prosječno iskorištenje vage =  $(B_V / N_V) / T$
- Prosječno iskorištenje punjača =  $(B_P / N_P) / T$

Po mojoj definiciji inicijalnih uslova postoje 3 punjača i dvije vage pa je :

- $N_P = 3$
- $N_V = 2$

Spomenute mjere performansi će biti obrađene u poglavlju 2 (Glavni dio rada) i 3 (Opis rezultata).

## 2. Glavni dio rada, Opis metode

### 2.1 Prepoznavanje glavnih komponenti simulacije

Entitet :

- 8 kamiona za prijevoz (TR1, TR2, ... , TR8)

Kako se radi o diskretnom broju kamiona, možemo zaključiti da se radi o konačnoj (eng. finite) populaciji (eng. calling population). Dalje na osnovu ovog zaključka možemo dalje zaključiti da je učestalost dolaska kamiona (eng. arrival rate) u sistem predvidiva ovisno o broju kamiona koji su trenutno u samom sistemu za čekanje (npr. Učestalost dolaska kamiona je maksimalna kada se u sistemu niti jedan kamiona ne nalazi).

Eventi :

- Punjenje kamiona (L)
- Vaganja (Q)
- Putovanje (P)

Stanja sistema:

- $LQ(t)$  – broj kamiona u redu za punjenje
- $L(t)$  – broj kamiona koji se puni (0, 1, 2, 3)
- $WQ(t)$  – broj kamiona u redu za vaganje
- $Q(t)$  – broj kamiona koji se važe (0, 1, 2)

Aktivnosti :

- vrijeme trajanja punjenja
- vrijeme trajanja vaganja
- vrijeme trajanja putovanja

Zastoj/kašnjenje :

- Kamion čeka u redu za punjenje
- Kamion čeka u redu za vaganje

Obavještenje događaja :

- $(ALQ, t, TRi)$  : predstavlja događaj dolaska kamiona i u LQ u budućem vremenu  $t$
- $(EL, t, TRi)$  : predstavlja događaj završavanja punjena kamiona i u budućem vremenu  $t$
- $(EW, t, TRi)$  : predstavlja događaj završavanja vaganja kamion i u budućem vremenu  $t$

Liste:

- LQ – red čekanja za punjenje (FCFS/FIFO)
- WQ – red čekanja za vaganje (FCFS/FIFO)

Ulazni podaci na kojima se vrši simulacije se dobivaju sa generatorima slučajnih varijabli po sljedećim distribucijama:

- Weibullova
- Uniformna

Za ulazne podatke je potrebno generisati sljedeća vremena:

- vrijeme punjenja (ovisi o broju zauzetih punjača, 3 max zauzeta punjača)
- vrijeme vaganja (ovisi o broju zauzetih vagi, 2 max zauzeta vagi)
- vrijeme putovanja (za jedan ciklus simulacije dovoljno je 8 vremena)

## 2.2 Vršenje simulacije

Kako bi se izvršilo generisanje slučajnih ja sam kreirao program u c++ sa inicijalnim skupom podataka R1 i R2 (eng. seed data), te koristio tehniku inverzne transformacije i prihvatanje-odbijanja (eng. acceptance-rejection) za generisanje slučajnih po sljedećim distribucijama :

- Poisson
- Weibull
- Triangular
- Uniforme
- Exponential

Rezultate sam promatrao kroz konzolni program i zaključio da su uniformna i weibullova dale najbolje rezultate i time odlučio da ih koristim za vršenje generisanja slučajnih varijacija. Prednosti ove dvije u odnosu na ostale je mogućnost odabira intervala iz kojeg želim da generišem slučajne a da pri tome rezultati budu na oko realni. Pošto moja simulacija ne radi sa beskonačnim skupom populacije za ulazak u sistem (eng. infinite calling population) poissonova distribucija baš i nema smisla za vremena trajanja putovanja.

Rezultati generisanja slučajnih varijacija po Weibullovoj i uniformnoj distribuciji:

TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8
30 min	29 min	31 min	31 min	30 min	29 min	30 min	29 min

**Tabela 1.** Vremena trajanja putovanja kamiona (Weibull distribucija)

TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8
11 min	10 min	17 min	18 min	18 min	18 min	19 min	12 min

**Tabela 2.** Vremena trajanja punjenja kamiona (Uniformna distribucija)

TR1	TR2	TR3	TR4	TR5	TR6	TR7	TR8
8 min	7 min	9 min	10 min	9 min	7 min	8 min	7 min

**Tabela 3.** Vremena trajanja vaganja kamiona (Uniformna distribucija)

### Simulacija korištenjem slučajnih iz tabela 1, 2 i 3

Clock t	LQ(t)	L(t)	WQ(t)	W(t)	Lista LQ	Lista WQ	Lista Future Event	B <sub>p</sub>	B <sub>v</sub>
0	5	3	0	0	TR8 TR7 TR6 TR5 TR4	-//-	(EL,10,TR2) (EL,11,TR1) (EL,17,TR3)	0	0
10	4	3	0	1	TR8 TR7 TR6 TR5	-//-	(EL,11,TR1) (EL,17,TR3) (EL,10+18,TR4) (EW,10+7,TR2)	30	0
11	3	3	0	2	TR8 TR7 TR6	-//-	(EL,17,TR3) (EL,28,TR4) (EL,18+11,TR5) (EW,17, TR2) (EW,11+8,TR1)	33	1
17	2	3	0	2	TR8 TR7	-//-	(EL,28,TR4) (EL,29,TR5) (EL,17+18,TR6) (EW,19,TR1) (EW,17+9,TR3) (ALQ, 17+29,TR2)	51	13



19	2	3	0	1	TR8 TR7	-//-	(EL,28,TR4) (EL,29,TR5) (EL,35,TR6) (EW,17+9,TR3) (ALQ, 19+30,TR1) (ALQ, 46,TR2)	57	17
26	2	3	0	0	TR8 TR7	-//-	(EL,28,TR4) (EL,29,TR5) (EL,35,TR6) (ALQ, 49, TR1) (ALQ, 46,TR2) (ALQ, 26+31,TR3)	78	24
28	1	3	0	1	TR8	-//-	(EL,29,TR5) (EL,35,TR6) (EL, 28+19, TR7) (EW,28+10,TR4) (ALQ, 49, TR1) (ALQ, 46,TR2) (ALQ, 57,TR3)	84	24
29	0	3	0	2	-//-	-//-	(EL,35,TR6) (EL,47, TR7) (EL,29+12,TR8) (EW,38,TR4) (EW,29+9,TR5) (ALQ,46,TR2) (ALQ,49, TR1) (ALQ,57,TR3)	87	25
35	0	2	1	2	-//-	TR6	(EL,47, TR7) (EL,41,TR8) (EW,38,TR4) (EW,38,TR5) (ALQ,46,TR2) (ALQ,49, TR1) (ALQ,57,TR3)	105	37
38	0	2	0	1	-//-	-//-	(EL,47, TR7) (EL,41,TR8) (EW,38+7,TR6) (ALQ,46,TR2) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,38+31,TR4) (ALQ,38+30,TR5)	111	43

41	0	1	0	2	-//-	-//-	(EL,47, TR7) (EW,45,TR6) (EW, 41+7,TR8) (ALQ,46,TR2) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4)	117	46
45	0	1	0	1	-//-	-//-	(EL,47, TR7) (EW, 48,TR8) (ALQ,46,TR2) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,42+29,TR6)	121	54
46	1	1	0	1	-//-	-//-	(EL,47, TR7) (EW, 48,TR8) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,42+29,TR6)	122	55
47	1	0	0	2	-//-	-//-	(EW,47+8, TR7) (EW, 48,TR8) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,71,TR6)	123	56
48	1	0	0	1	-//-	-//-	(EW,55, TR7) (ALQ,48+29,TR8) (ALQ,49, TR1) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,71,TR6)	123	58

49	2	0	0	1	-//-	-//-	(EW,55, TR7) (ALQ,48+29,TR8) (ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,71,TR6)	123	59
55	2	0	0	0	-//-	-//-	(ALQ,57,TR3) (ALQ,68,TR5) (ALQ, 69,TR4) (ALQ,71,TR6) (ALQ,77,TR8) (ALQ,85,TR7)	123	65

Kako bi se izvršila validacija simulacije predstavljene u tabeli iznad korišten je programski jezik c++.

Radi se o template funkciji koja postavlja inicijalne podatke, vrši simulaciju i daje izvještaj o rezultatima simulacije korištenjem istih ulaznih podataka iz simulacije iznad (eng. random variate generators). Korisnost simulacije se ogleda u njenoj modifikabilnosti po broju servera, redova, konačnih/beskonačnih kapaciteta i calling populacija (generisanje podataka može se otežati za veće brojeve ciklusa).

```

-----Simulation has started-----
-----You can view the events bellow-----
Truck 1 has started loading at 0 minutes.
Truck 2 has started loading at 0 minutes.
Truck 3 has started loading at 0 minutes.
Truck 4 has started loading at 10 minutes.
Truck 2 has finished loading at 10 minutes.
Truck 2 has started weighting at 10 minutes.
Truck 5 has started loading at 11 minutes.
Truck 1 has finished loading at 11 minutes.
Truck 1 has started weighting at 11 minutes.
Truck 6 has started loading at 17 minutes.
Truck 3 has finished loading at 17 minutes.
Truck 2 has finished weighting at 17 minutes.
Truck 3 has started weighting at 17 minutes.
Truck 1 has finished weighting at 19 minutes.
Truck 3 has finished weighting at 26 minutes.
Truck 7 has started loading at 28 minutes.
Truck 4 has finished loading at 28 minutes.
Truck 4 has started weighting at 28 minutes.
Truck 8 has started loading at 29 minutes.
Truck 5 has finished loading at 29 minutes.
Truck 5 has started weighting at 29 minutes.

```

Prikaz nekoliko događaja

```

----- Report -----
Cumulativ loading time -> 123
Cumulativ weighting time -> 65
-----
Coefficient of utilization of weighters -> 0.5818
Coefficient of utilization of loaders -> 0.7455
-----
Loading time :
Mean -> 15.3562
Variance -> 10.9460
Std. Dev. -> 3.3085
-----
Weighting time :
Mean -> 8.1650
Variance -> 2.6751
Std. Dev. -> 1.6356
-----
Traveling time :
Mean -> 29.9305
Variance -> 0.9697
Std. Dev. -> 0.9847

```

Izvještaj simulacije

## 2.3 Opis Metode

Korištenjem uniformne i weibullove distribucije uspješno je se izvršila simulacija. Tehnika inverzne transformacije na CDF spomenutih distribucija ja dala zadovoljavajuće rezultate. Opis generisanje slučajnih tehnikom inverzne transformacije (Weibull & Uniformna).

Za Weibull Distribuciju :

**Korak 1.** Prepoznavanje CDF ->  $F(x) = 1 - e^{-(x/\alpha)^\beta}$ ,  $x \geq 0$

**Korak 2.** Postaviti da je  $R = F(X) = 1 - e^{-(x/\alpha)^\beta}$

**Korak 3.** Rješiti X za R prema inverznoj jednačini  $X = \alpha[-\ln(1-R)]^{1/\beta}$

Spomenute korake iznad sam preveo u sljedeći programski kod :

```
float* weibull_rand_gen(float alfa, float beta, float R[], int size) {
    float* result = NULL;
    result = new float[size];
    for (int i = 0; i < size; i++)
    {
        float logarithm = log(1 - R[i]);
        float exponent = 1 / beta;
        result[i] = alfa * pow(-logarithm, exponent);
    }
    return result;
}
```

Za Uniformnu Distribuciju :

**Korak 1.** Prepoznavanje CDF -> 
$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$$

**Korak 2.** Postaviti da je  $R = F(X) = (X-a)/(b-a)$

**Korak 3.** Rješiti X za R prema inverznoj jednačini  $X = a + (b-a)R$

Spomenute korake iznad sam preveo u sljedeći programski kod :

```
float* uniform_rand_gen(float a, float b, float *R, int size) {
    float* result = NULL;
    result = new float[size];
    for (int i = 0; i < size; i++)
    {
        result[i] = a + (b - a) * R[i];
    }
    return result;
}
```

### 3. Opis rezultata, dijagrami

Kao što sam spomenuo cilj simulacije je bilo napraviti estimaciju iskorištenja vage i punjača kao glavna mjera performansi. Vremena zauzetosti vagi i punjača se mogu vizualizirati histogramima (štapićastim dijagramima) .

Spomenute mjere performansi će se računati za jedan puni ciklus trajanja simulacije, gdje se jednim ciklusom smatra jedan obilazak kamiona kroz čitav sistem. Više ciklusa bi dalo bolje rezultate ali bi zauzelo i nekoliko desetina listova kako bi se prikazala simulacija tabelarno što je nepraktično.

#### 3.1 Estimacija iskorištenja vagi (Glavna mjera performansi)

Prije početka izračuna prepoznati ću nekoliko bitnih podataka :

- $N_P = 3$  (broj punjača)
- $N_V = 2$  (broj vagi)
- $B_P = 138$  (ukupno vrijeme zauzetosti punjača)
- $B_V = 62$  (ukupno vrijeme zauzetosti vagi)
- $T = 55$  (ukupno vrijeme trajanja simulacije za jedan ciklus)

Formule po kojima ću računati mjeru performansi iskorištenja vagi i punjača (eng. servers) :

- Prosječno iskorištenje vagi =  $(B_V / N_V) / T$
- Prosječno iskorištenje punjača =  $(B_P / N_P) / T$

Po formuli i podacima iznad vršim estimaciju iskorištenja servera :

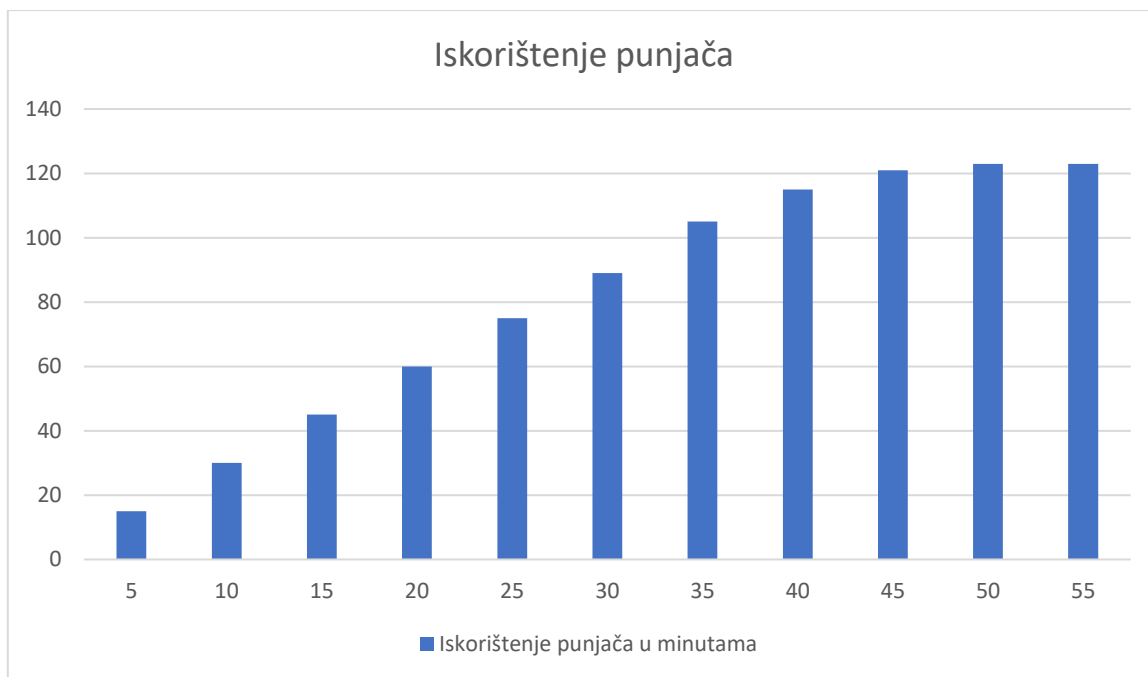
$$\text{Prosječno iskorištenje vagi} = (65/2)/55 = 0.581$$

$$\text{Prosječno iskorištenje punjača} = (123/3)/55 = 0.745$$

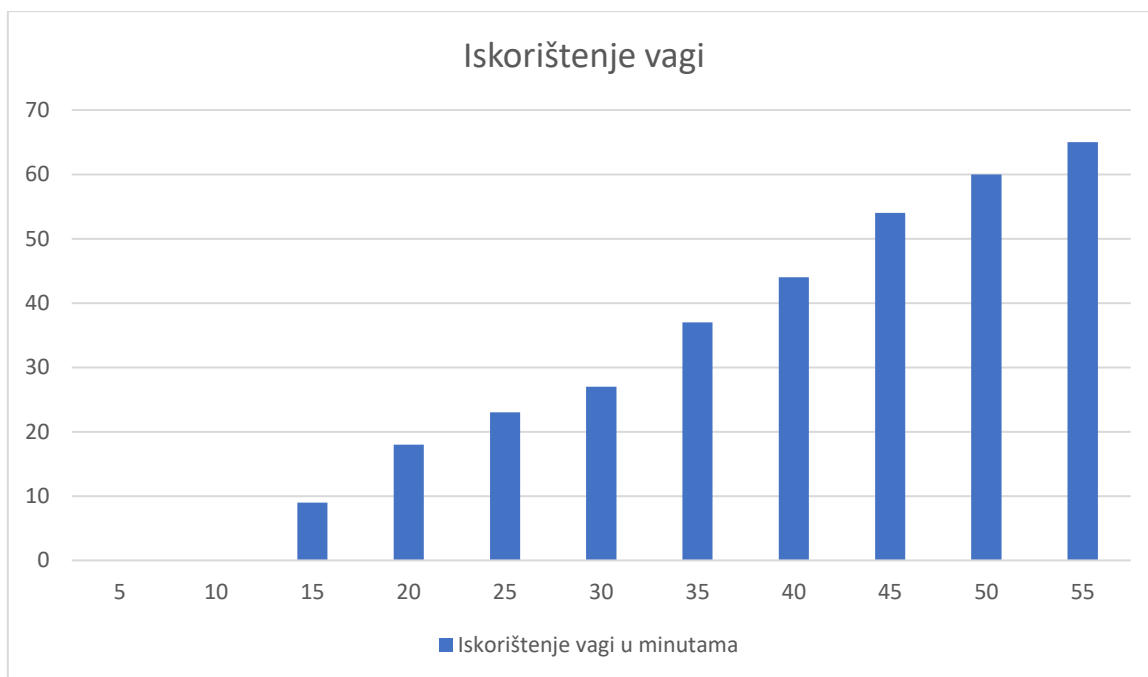
Loading time :	Weighting time :	Traveling time :
Mean -> 15.3562	Mean -> 8.1650	Mean -> 29.9305
Variance -> 10.9460	Variance -> 2.6751	Variance -> 0.9697
Std. Dev. -> 3.3085	Std. Dev. -> 1.6356	Std. Dev. -> 0.9847

Program ispisuje i drugi korisni pokazatelji kao sredina, varjansa i standardna devijacija.

## 3.2 Dijagrami



Dijagram 1. iskorištenja punjača



Dijagram 2. Iskorištenja vagi

Dijagrami iznad prikazuju kumulativnu sumu iskorištenja server u intervalima po 5 minuta.

## 4. Budući rad

Kako su vage i punjači ne iskorišteni do svoga maksimuma, moguće je promijeniti broj vagi i punjača kao i redove kako bi se našla najoptimalnija kombinacija sa ciljem maksimizacije iskorištenja punjača i vagi.

Promjena brojeva vagi, punjača i redova u tabelarnoj simulaciji je lagano ali vršenje same simulacije za različite kombinacije bi uzelo mnogo vremena. Pored vršenja simulacije generisanje slučajnih ručno koje simulacije prima kao ulaz bi bio ogroman problem (do nekoliko stotina pa i hiljada slučajnih).

Moje ideja za budući rad je kreirati simulacijski program sa konfigurabilnim brojem redova, vagi i punjača. Čak jedan korak u naprijed bi bio definisanje takve strukture programa da podržava bilo koji queueing sistem. Moj originalni program namijenjen za 3 punjača, 2 vage i 2 reda za svaku vrstu posla bi se mogla modifikovati kako bi podržavala različiti vrste calling populacije (finiti/inifinite), redova (finiti/infinite kapacitet, single/multi queue) i naravno različite vrste servera (paralelno/serijski vezani, single/multi server).

Pošto simulacioni program zahtjeva ulazne podatke (slučajne), moguće je iskoristiti moj postojeći random variate generator. Također moji generatori podržavaju 6 distribucija, pa bi se broj distribucija mogao povećati, a kao ulazni parametri za generatore (eng. seed data) bi se mogli koristiti linearni kongruentni generatori. Kako bi na kraju bili sigurni da je simulacije tačna mogla bi se izvesti testiranje slučajnih na tkz. randomness korištenjem testova kao Kolmogorv-Smirnov ili Chi-squared test.

Sa ovim bi bilo moguće izvesti stotine simulacija sa hiljadama slučajnih kao ulazi i dužine od stotine ciklusa (veći broj ciklusa, slučajnih i simulacije bi zahtijevalo snažnije komputacije od trajanja dužih vremenskih perioda).

Opcije unaprjeđenja simulacionog programa :

- Definisanje generičke strukture koja podržava sve queueing sisteme
- Proširenje broja distribucija koje generator može da koristi
- Korištenje linearnih kongruentnih generatora za kreiranje seeding podataka
- Testiranje seeding podataka na randomness sa Kolmogorov-Smirnov i Chi-squared testovima
- Uvođenje budžeta i cijena komponenti sistema

Rezultat vršenja modifikacije programa bi rezultirao u ponovnoj upotrebljivosti za različite poslovne domene kao alat za optimiziranje mjera performansi njihovog sistema.

## 5. Zaključak

Na osnovu simulacije se može zaključiti da broj vagi i punjača nisu optimalni pa bi kompanija za dostavu uglja mogla promijeniti broj vagi i punjača. Glavni uzrok ne iskorištenja vagi i punjača je bottleneck koji postoji na samim mjestima obrade kamiona (vagama i punjačima).

Iskorištenje vagi i punjača :

$$\text{Prosječno iskorištenje vagi} = (65/2)/55 = 0.581$$

$$\text{Prosječno iskorištenje punjača} = (123/3)/55 = 0.745$$

Na osnovu koeficijenata odbijenih nakon vršenja simulacije može se reći da vage najviše trpe od bottleneck-a na punjačima. Pošto je koeficijent vagi približno 50% može se zaključiti da je se većinom samo jedna vaga koristila. Ovisno o potrebama kompanije ta vaga se može i iskorištenje bi se popelo na 100%. Također 33% je maksimalna vrijednost iskorištenja koju svaki od punjača može da ima u donosu na ostala 2 punjača. Kako je koeficijent iskorištenja približno 75% može se zaključiti da su 2 maksimalno iskorištena dok je 3. punjač iskorišten 9% a neiskorišten 23% (rezultati su dobiveni običnom aritmetikom). Dakle ako je cilj kompanije da se poveća iskorištenje punjača mogao bi se jedan ukloniti kako je  $9\% < 23\%$  (da je iskorištenje 3. punjača bilo 50%, odnosno 16.5% mogao bi se ostaviti).

Sami redovi nemaju utjecaja na koeficijent iskorištenja servera jer se unutar njih može nalaziti 8 kamiona istovremeno. Smanjenjem kapaciteta redova bi se napravio bottleneck, posebice na redu za čekanja na punjenje. Na kraju zaključujem da bi iskorištenje vagi i punjača bio najoptimalniji za 8 punjača i 8 vagi, ali to je najskuplja opcija za kompaniju za dostavu uglja i ne bih je preporučio.

Ukoliko koristimo simulacioni softver za određivanje najoptimalnijih mjera performansi kao iskorištenje servera mogle bi se uvesti nove varijable kao budžet i cijena vagi, punjača i redova. Simulacioni softver bi trebao da isproba sve moguće kombinacije po broju i kapacitetu punjača, vagi i redova poštujući cijene servera i budžet. Rezultat bi trebala da bude najoptimalnija kombinacija punjača, vagi i redova u okviru kompanijinog budžeta za proširenje sistema.



## 6. Literatura

[1] Discrete-event system simulation 4th edition (Jerry banks, Johns S. Carson, Barry L. Nelson, David M. Nicol)

[2] Wikipedia

[3] Indian Institute of Technology Roorke Youtube channel