

SVEUČILIŠTE U MOSTARU
FAKULTET STROJARSTVA, RAČUNARSTVA I ELEKTROTEHNIKE

SEMINARSKI RAD
AUTOPRAONICA

Ivan Đolo
Dragan Zovko

Mostar, srpanj 2024.

SADRŽAJ

1	Uvod	1
2	Opis problema	2
3	Metodologija	3
3.1	Korišteni alati i tehnologije	3
3.2	Definiranje agenata	3
3.3	Parametri simulacije.....	3
3.4	Raspored dolazaka klijenata	4
3.5	Model simulacije.....	4
4	Prikaz koda i funkcionalnosti simulacije	5
5	Zaključak.....	14

POPIS SLIKA

Slika 4.1. Deklaracija vrsta i varijabli	5
Slika 4.2. Inicijalizacija modela simulacije	6
Slika 4.3. Glavna petlja simulacije	6
Slika 4.4. Ažuriranje sata simulacije	7
Slika 4.5. Inicijalizacija perača	7
Slika 4.6. Dinamičko ažuriranje perača	8
Slika 4.7. Postavljanje mjesta za pranje	8
Slika 4.8. Generiranje automobila	9
Slika 4.9. Procesiranje automobila	10
Slika 4.10. Ažuriranje perača	11
Slika 4.11. Ažuriranje troškova	12
Slika 4.12. Završetak radnog dana i spremanje rezultata	12
Slika 4.13. Prikaz simulacijskog okruženja	13

1 UVOD

Autopraonica je svakodnevni dio života koji osigurava čistoću i održavanje vozila. Upravljanje takvim objektima može biti izazovno zbog potrebe za optimizacijom radne snage i resursa kako bi se osigurala učinkovita usluga i maksimizirala zarada. U ovom radu opisujemo simulaciju rada autopraonice pomoću NetLogo programskog jezika. Cilj je razviti model koji će nam omogućiti testiranje različitih strategija zapošljavanja perača automobila kako bismo optimizirali rad autopraonice uz istovremeno održavanje visoke kvalitete usluge i smanjenje vremena čekanja za klijente.

2 OPIS PROBLEMA

Autopraonica se suočava s nizom izazova u svakodnevnom poslovanju. Primarni cilj autopraonice je osigurati brz i kvalitetan servis uz istovremeno maksimiziranje zarade. Ključni izazovi uključuju:

- **Optimizacija radne snage:** Određivanje optimalnog broja perača automobila potrebnih za obradu dolazaka klijenata u autopraonicu.
- **Upravljanje vremenom:** Osiguranje da se klijenti poslužuju u razumnom vremenskom roku kako bi se smanjio broj klijenata koji odustaju zbog dugog čekanja.
- **Maksimiziranje zarade:** Pronalaženje ravnoteže između troškova rada perača automobila i prihoda od usluga kako bi se osigurala profitabilnost poslovanja.

3 METODOLOGIJA

3.1 Korišteni alati i tehnologije

Za simulaciju rada autopraonice koristili smo NetLogo alat za modeliranje i simulaciju složenih sustava pomoću agenata. NetLogo pruža jednostavno sučelje za modeliranje agenata i omogućuje simulaciju složenih scenarija i interakciju između agenata.

3.2 Definiranje agenata

U našoj simulaciji definirali smo dvije glavne vrste agenata:

- **Perači automobila:** Odgovorni za pružanje usluga pranja automobila. Svaki perlač ima određene vještine i može opslužiti klijente u određenom vremenskom okviru.
- **Klijenti:** Predstavljaju osobe koje dolaze na uslugu. Vrijeme potrebno za uslugu je 15 minuta za osobne automobile, 20 minuta za kombi vozila i 30 minuta za teretna vozila.

3.3 Parametri simulacije

Ovo su parametri naše simulacije:

- Plaća perlača: 30 KM po satu
- Vrijeme trajanja usluge: 15, 20 i 30 minuta
- Cijena usluge: 20 KM
- Odustajanje klijenata: Klijenti koji čekaju duže od 30 minuta odustaju od usluge

3.4 Raspored dolazaka klijenata

Simulirali smo dolazak klijenata u autopraonicu tijekom radnog dana u različitim vremenskim intervalima:

- Od 08:00 do 10:00 - 20 klijenata po satu
- Od 10:00 do 14:00 - 30 klijenata po satu
- Od 14:00 do 18:00 - 25 klijenata po satu

3.5 Model simulacije

Naš model simulacije obuhvaća nekoliko koraka specifičnih za rad autopraonice:

- **Inicijalizacija:** Postavljanje početnih uvjeta za simulaciju uključujući stvaranje peraa automobila i definiranje ulaza i izlaza za klijente.
- **Dodjela klijenata peraa:** Klijenti se dodjeljuju slobodnim peraa na temelju dostupnosti.
- **Upravljanje redovima ekanja:** Klijenti eekaju na uslugu, a ako eekanje traje duže od 30 minuta, odlaze iz autopraonice.
- **Praćenje zarade i troškova:** Evidentiranje zarade od usluga i troškova rada peraa.
- **Optimizacija broja peraa:** Simulacija automatski dodaje nove peraa kada je potražnja visoka i uklanja ih kada više nisu potrebni.

4 PRIKAZ KODA I FUNKCIONALNOSTI SIMULACIJE

Slika 4.1. ovaj dio koda deklarira dvije vrste agenata: washers (perači) i cars (automobili). Globalne varijable uključuju earnings (zarada), costs (troškovi), total-cars (ukupni automobili), satisfied-cars (zadovoljni automobili), total-waiting-time (ukupno vrijeme čekanja), hours (sati), minutes (minute), profit (profit) i success-rate (stopa uspješnosti). Varijable washers-own i cars-own definiraju specifične atribute za perače i automobile.

```
1 breed [washers washer]
2 breed [cars car]
3
4 globals [
5   earnings
6   costs
7   total-cars
8   satisfied-cars
9   total-waiting-time
10  hours
11  minutes
12  profit
13  success-rate
14 ]
15
16 washers-own [
17   is-washing?
18   car-being-washed
19 ]
20
21 cars-own [
22   waiting-time
23   wash-duration
24   car-type ; 1 for personal, 2 for van, 3 for truck
25   wash-spot ; 1 for gray (personal/vans), 2 for black (vans/trucks)
26 ]
27
```

Slika 4.1. Deklaracija vrsta i varijabli

Slika 4.2. setup inicijalizira simulaciju, čisteći sve prethodne postavke (clear-all), resetira broj otkucaja (reset-ticks) i postavlja početne vrijednosti globalnih varijabli. Funkcije initialize-washers i setup-wash-spots postavljaju peraae i definiraju mjesta za pranje automobila.

```
28  to setup
29    clear-all
30    reset-ticks
31    set earnings 0
32    set costs 0
33    set total-cars 0
34    set satisfied-cars 0
35    set total-waiting-time 0
36    set hours 8
37    set minutes 0
38    initialize-washers
39    setup-wash-spots
40  end
```

Slika 4.2. Incijalizacija modela simulacije

```
42  to go
43    if ticks > 600 [
44      end-of-day
45      stop
46    ]
47
48    tick
49    update-clock
50    update-washers-dynamically
51    update-washers
52    generate-cars
53    process-cars
54    update-costs
55  end
56
```

Slika 4.3. Glavna petlja simulacije

Na slici 4.3 go je glavna petlja koja upravlja simulacijom. Petlja se zaustavlja nakon 600 otkucaja (što predstavlja 10 radnih sati). Unutar petlje se ažurira sat (update-clock), dinamički se prilagođava broj peraa (update-washers-dynamically), ažurira status peraa (update-washers), generiraju novi automobili (generate-cars), procesuiraju automobili (process-cars) i ažuriraju troškovi (update-costs).

```

80 ;; Update the simulation clock
81 to update-clock
82   set minutes minutes + 1
83   if minutes >= 60 [
84     set minutes 0
85     set hours hours + 1
86   ]
87 end

```

Slika 4.4. Ažuriranje sata simulacije

Funkcija update-clock ažurira minute simulacije. Kada minute dosegnu 60, resetiraju se na 0 i povećavaju se sati za 1.

```

65 to initialize-washers
66   create-washers num-washers [
67     set color blue
68     set shape "person"
69     set is-washing? false
70     set car-being-washed nobody
71     position-washer who
72   ]
73 end
74
75 to position-washer [index]
76   ifelse index < 2 [
77     let wxcor min (list max-pxcor (max (list min-pxcor ((index mod 2) * 2 + max-pxcor / 2))))
78     let wycor min (list max-pycor (max (list min-pycor (floor (index / 2) * 3))))
79     setxy wxcor wycor
80   ]
81   let wxcor min (list max-pxcor (max (list min-pxcor ((index mod 2) * 2 - 1))))
82   let wycor min (list max-pycor (max (list min-pycor (floor ((index - 2) / 2) * 3))))
83   setxy wxcor wycor
84 ]
85 end

```

Slika 4.5. Inicijalizacija perača

Na slici 4.5 funkcija initialize-washers stvara agente perače, postavlja njihove atribute i pozicije pomoću funkcije position-washer. position-washer određuje početne koordinate perača na temelju njihovog indeksa.

```

87 to update-washers-dynamically
88   let current-washers count washers
89   if current-washers < num-washers [
90     let new-washers num-washers - current-washers
91     create-washers new-washers [
92       set color blue
93       set shape "person"
94       set is-washing? false
95       set car-being-washed nobody
96       let unpaired-washers washers with [not any? other washers with [is-washing? and car-being-washed = nobody]]
97       if any? unpaired-washers [
98         let partner one-of unpaired-washers
99         setxy ([xcor] of partner + 1) ([ycor] of partner)
100       ]
101       position-washer who
102     ]
103   ]
104   if current-washers > num-washers [
105     let excess-washers current-washers - num-washers
106     let unpaired-washers washers with [not any? other washers with [is-washing? and car-being-washed = nobody]]
107     ifelse count unpaired-washers >= excess-washers [
108       ask n-of excess-washers unpaired-washers [
109         die
110       ]
111     ] [
112       ask n-of (excess-washers - count unpaired-washers) washers with [is-washing? = false] [
113         die
114       ]
115       ask unpaired-washers [
116         die
117       ]
118     ]
119   ]
120   ;; Ensure correct positioning after adding or removing washers
121   ask washers [
122     position-washer who
123   ]
124 end
125 ...

```

Slika 4.6. Dinamičko ažuriranje peraa

Na slici 4.6 funkcija `update-washers-dynamically` dinamički prilagođava broj peraa u simulaciji na temelju trenutnog broja peraa. Ako je broj peraa manji od željenog broja, stvaraju se novi peraa. Ako je broj peraa veći, višak peraa se uklanja. Nakon dodavanja ili uklanjanja peraa, svi peraa se ponovno pozicioniraju.

```

126 to setup-wash-spots
127   ask patches with [pxcor < max-pxcor / 2] [
128     set pcolor gray ; Spot for personal and vans
129   ]
130   ask patches with [pxcor >= max-pxcor / 2] [
131     set pcolor black ; Spot for vans and trucks
132   ]
133 end

```

Slika 4.7. Postavljanje mjesta za pranje

Na slici 4.7 funkcija `setup-wash-spots` postavlja boje zakrpa u radnom prostoru. Zakrpe s lijeve strane su sive i predstavljaju mjesta za pranje osobnih automobila i kombija, dok su zakrpe s desne strane crne i predstavljaju mjesta za pranje kombija i kamiona.

```

135 to generate-cars
136   let new-cars 0
137
138   if hours >= 8 and hours < 10 [
139     set new-cars random-poisson 0.33333 ;20/60
140   ]
141   if hours >= 10 and hours < 14 [
142     set new-cars random-poisson 0.5 ;30/60
143   ]
144   if hours >= 14 and hours < 18 [
145     set new-cars random-poisson 0.41666 ;25/60
146   ]
147
148   create-cars new-cars [
149     set color red
150     set waiting-time 0
151     set wash-duration choose-wash-duration
152     set car-type choose-car-type
153     ifelse (car-type = 1 or car-type = 2) [
154       set wash-spot 1
155     ] [
156       set wash-spot 2
157     ]
158     setxy random-xcor random-ycor
159     set total-cars total-cars + 1
160   ]
161 end
162
163 to-report choose-wash-duration
164   let r random-float 1
165   if r < 0.5 [ report 15 ]
166   if r >= 0.5 and r < 0.85 [ report 20 ]
167   report 30
168 end
169
170 to-report choose-car-type
171   let r random-float 1
172   if r < 0.5 [ report 1 ] ; Personal
173   if r >= 0.5 and r < 0.85 [ report 2 ] ; Van
174   report 3 ; Truck
175 end

```

Slika 4.8. Generiranje automobila

Na slici 4.8. funkcija generate-cars generira nove automobile na temelju trenutnog vremena (sati). Broj automobila generiranih u svakom satu određuje se Poissonovom raspodjelom. Svaki novi automobil dobiva atribute kao što su waiting-time, wash-duration, car-type i wash-spot. Funkcije choose-wash-duration i choose-car-type vraćaju nasumično trajanje pranja i vrstu automobila.

```

177 to process-cars
178   ask cars [
179     set waiting-time waiting-time + 1
180
181     ;; Leave if waiting too long
182     if waiting-time > 30 [
183       die
184     ]
185
186     ;; Find available washer pairs
187     let available-washers washers with [is-washing? = false]
188     if count available-washers >= 2 [
189       let target-washers n-of 2 available-washers
190       if count target-washers = 2 [
191         let target-x mean [xcor] of target-washers
192         let target-y mean [ycor] of target-washers
193         move-to-target target-x target-y
194         if distancexy target-x target-y < 2 [
195           ask target-washers [
196             set is-washing? true
197             set car-being-washed myself
198           ]
199           set color white
200         ]
201       ]
202     ]
203   ]
204 end
205
206 to move-to-target [target-x target-y]
207   face patch target-x target-y
208   fd 1
209 end

```

Slika 4.9. Procesiranje automobila

Slika 4.9. funkcija process-cars upravlja ponašanjem automobila. Automobili povećavaju svoje vrijeme čekanja svake minute. Ako čekaju duže od 30 minuta, napuštaju praonicu. Automobili traže dostupne parove perača i ako ih pronadu, pomiču se prema njima i započinju proces pranja. Funkcija move-to-target omogućava automobilima da se pomiču prema ciljanom mjestu.

```

211  to update-washers
212    ask washers [
213      if is-washing? [
214        if car-being-washed != nobody [
215          ask car-being-washed [
216            set wash-duration wash-duration - 1
217            if wash-duration <= 0 [
218              set earnings earnings + 20
219              set satisfied-cars satisfied-cars + 1
220              set total-waiting-time total-waiting-time + waiting-time
221              die
222            ]
223          ]
224        ]
225        if car-being-washed = nobody [
226          set is-washing? false
227        ]
228      ]
229    ]
230  end
231

```

Slika 4.10. Ažuriranje peraća

Na slici 4.10. funkcija update-washers upravlja ponašanjem peraća tijekom pranja automobila. Ako perać pere automobil, smanjuje se vrijeme pranja automobila. Kada je pranje završeno, automobil odlazi, zarada se povećava za 20 KM, a broj zadovoljnih automobila i ukupno vrijeme čekanja se ažuriraju.

```

232 to update-costs
233   let active-washers count washers with [is-washing? = true]
234   set costs costs + (active-washers * 30 / 60) ; Costs are hourly wage per minute
235 end
236

```

Slika 4.11. Ažuriranje troškova

Funkcija update-costs ažurira troškove rada peraa. Troškovi se izračunavaju na temelju broja aktivnih peraa i njihove satnice, preračunate po minuti.

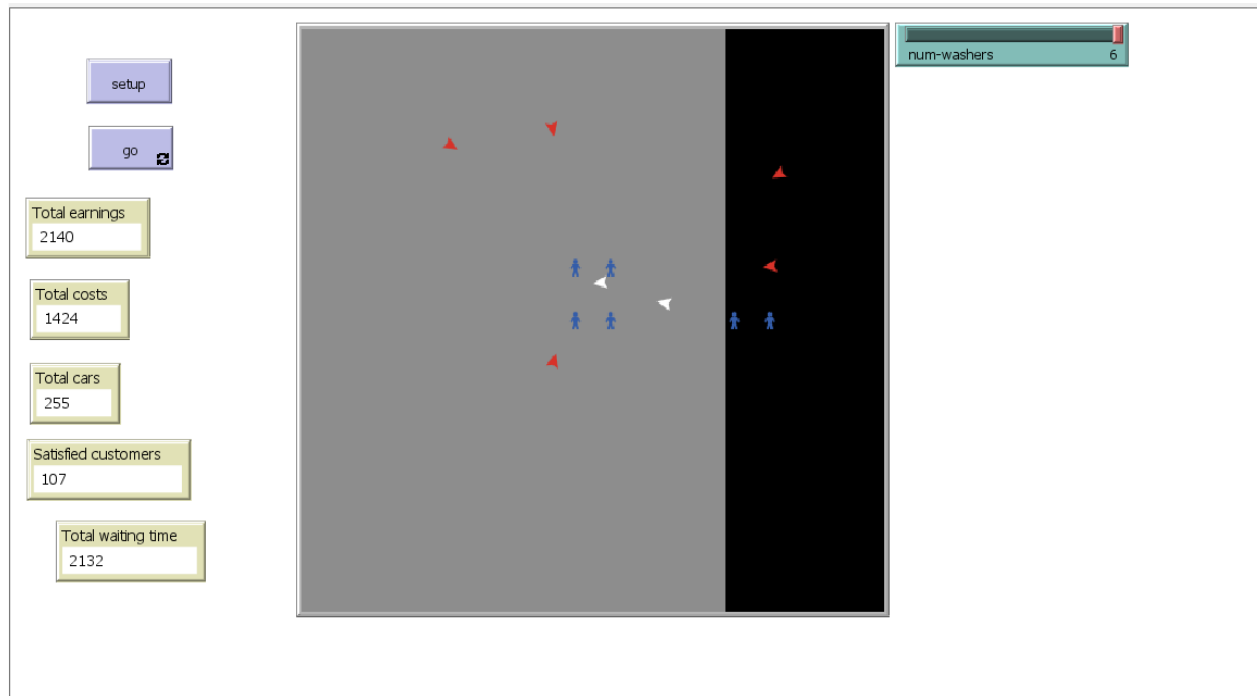
```

237 to end-of-day
238   calculate-profit-and-success
239   save-results-to-excel
240 end
241
242 to calculate-profit-and-success
243   set profit earnings - costs
244   if costs > 0 [
245     set success-rate (earnings / costs) * 100
246   ]
247 end
248
249 to save-results-to-excel
250   let rounded-success-rate precision success-rate 3
251   let file-path (word user-directory "car_wash_results.csv")
252   output-print (word "Results saved to: " file-path) ;; Print the file path to the console
253   file-open file-path
254   file-print (word "Ukupna dnevna zarada: " earnings)
255   file-print (word "Ukupni dnevni troskovi: " costs)
256   file-print (word "Ukupni profit: " profit)
257   file-print (word "Uspjesnost: " rounded-success-rate "%")
258   file-print (word "Ukupno automobila: " total-cars)
259   file-print (word "Zadovoljni automobili: " satisfied-cars)
260
261   let avg-waiting-time "N/A"
262   if satisfied-cars > 0 [
263     set avg-waiting-time total-waiting-time / satisfied-cars
264   ]
265   file-print (word "Prosječno vrijeme čekanja: " avg-waiting-time)
266   file-print (word "\n")
267   file-close
268 end
269

```

Slika 4.12. Završetak radnog dana i spremanje rezultata

Funkcija end-of-day obrađuje kraj radnog dana, izračunava profit i stopu uspješnosti te sprema rezultate simulacije u CSV datoteku. calculate-profit-and-success izračunava profit kao razliku između zarade i troškova te izračunava stopu uspješnosti. save-results-to-excel sprema rezultate simulacije u datoteku, uključujući ukupnu dnevnu zaradu, troškove, profit, stopu uspješnosti, ukupan broj automobila, broj zadovoljnih automobila i prosječno vrijeme čekanja.



Slika 4.13. Prikaz simulacijskog okruženja

5 ZAKLJUČAK

Analizom simulacijskog modela autopraonice dobivamo uvid u optimalne strategije upravljanja brojem peraa automobila i rasporedom dolazaka klijenata. Korištenjem simulacijskog okruženja možemo testirati različite scenarije i postavke kako bismo povećali profitabilnost i učinkovitost autopraonice. Na temelju rezultata simulacije optimalno rješenje uključuje balansiranje broja peraa i rasporeda dolazaka klijenata kako bi se minimiziralo vrijeme čekanja i maksimizirala zarada autopraonice.