Univerza v Ljubljani
Fakulteta za matematiko in fiziko

THE POWER OF ADJACENT CHOICES

Finančni praktikum

Avtorja:
Justin Raišp, Maša Orelj

Ljubljana, 2022

1 Navodilo

Imamo n žog in n košev $b_1, \ldots, b_n$, ki so prazni. Koši so postavljeni v krogu: koš b_i ima soseda b_{i-1} in b_{i+1}, kjer zaradi krožnosti velja, da sta b_1 in b_n soseda. Opazujemo naslednji naključen proces: Za vsako žogo naključno izberemo koš b_i, pogledamo še oba soseda in damo žogo v koš z najmanjšim številom žog v tistem trenutku izmed teh treh. Zanima nas število žog v košu z največ žogami na koncu procesa. Ta proces potem razširimo na več načinov:

- za sosede štejemo koše, ki so na razdaljah največ $2, 3, \ldots,$
- za n košev vzamemo $2n, 3n, 4n, \ldots$ žog,
- iščemo koš z najmanjšim številom žog.

Opazujemo lahko tudi dvodimenzionalno mrežo košev s topologijo torusa. Torej imamo npr. n^2 košev $b_{i,j}$, kjer sta $i, j \in [n]$, kjer za soseda $b_{i,j}$ in $b_{k,l}$ velja $|i-k| + |j-l| = 1$. Soseda sta tudi $b_{1,i}$ in $b_{n,i}$ ter $b_{i,1}$ in $b_{i,n}$. Podobno lahko gledamo tudi tridimenzionalno verzijo.

2 Načrt dela

Za reševanje opisanega problema bova uporabila programski jezik *Python*. Najprej bova napisala program, s katerim bova lahko preizkušala vpliv spreminjanja števila sosedov in količine žog na poskus. Pri tem bova privzela, da v primeru, ko je košev z najmanjšim številom žog med pregledanimi koši več, položimo žogo v koš z najmanjšim indeksom. Za začetek sva pripravila prvo različico programa, ki je namenjena najbolj osnovnemu primeru poskusa:

$$\text{število košev} = n, \ \text{število žog} = n, \ \text{število levih oz. desnih sosedov} = 1.$$

```
#OSNOVNI PROGRAM
def najdi_sosede_1d(kosi,izbrana_kosarica, razdalja=1):
    sosedi = {}
    st_kosev = len(kosi)
    for j in range((-razdalja),(razdalja+1)):
        indeks_soseda = (izbrana_kosarica + j) % st_kosev #najde indeks soseda v list
        sosedi[indeks_soseda] = kosi[indeks_soseda] #v slovar sosedov dodamo njegov i
    min_vrednost = min(sosedi.values()) #izmed sosedov poiscemo minimalno vrednost
    return sosedi, min_vrednost

def maksimalno_stevilo_zogic(st_zogic, st_kosev, razdalja=1):
    zacetek = time.time()
    kosi = [0]*st_kosev
    for i in range(st_zogic):
        izbrana_kosarica = random.randrange(st_kosev)
        sosedi, min_vrednost = najdi_sosede_1d(kosi, izbrana_kosarica, razdalja) #iz
        kandidati = [k for k,v in sosedi.items() if v == min_vrednost] #poiscemo vse
        min_sosed = min(kandidati) #izmed kandidatov vzamemo tistega z najnizjim inde
        kosi[min_sosed] += 1 #minimalnemu sosedu dodamo zogo
    maksimum = max(kosi) #poiscemo maksimalno stevilo zog v posameznem kosu
    st_kosev_z_max_vrednostjo = kosi.count(max(kosi)) #prestejemo stevilo kosev, ki i
    delez_kosev = st_kosev_z_max_vrednostjo / st_kosev #izracunamo delez kosev z maks
    konec = time.time()
    casovna_zahtevnost = f'{konec-zacetek}' #izracunamo casovno zahtevnost algoritma
    return maksimum, casovna_zahtevnost, delez_kosev
```

V naslednjem koraku bova program prilagodila tako, da bova število sosedov in večkratnost žog vključila kot spremenljivki, na koncu pa ga bova prevedla še na dve oz. tri dimenzije.

3 Cilj

Cilj najne seminarske naloge je preučevati vpliv različnih spremenljivk (število sosedov, večkratnost žog) na maksimalno število žog v košari, ko so te razporejene v krogu, v torusu ali v treh dimezijah. Predstavljeni problem je problem teorije verjetnosti in pričakujeva, da bova lahko maksimalno število žog, v odvisnosti od omenjenih parametrov, napovedala z verjetnostjo blizu 1.