Univerza v Ljubljani
Fakulteta za matematiko in fiziko

THE POWER OF ADJACENT CHOICES

Finančni praktikum

Maša Orelj, Justin Raišp

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:

število košev $= n$, število žog $= n$, število levih oz. desnih sosedov $= 1$.

```
def maksimalno_stevilo_zogic(n):
    B = [0]*n
    x = 0
    while x <= n-1:
        izbrana_kosarica = random.randrange(n)
        if izbrana_kosarica == 0:
            sosedi = [B[-1]] + B[0:2]
            minimalni_sosed = min(sosedi)
            indeks = sosedi.index(minimalni_sosed)
            if indeks == 0:
                B[-1] += 1
            else:
                B[izbrana_kosarica -1] += 1
            x += 1
```

```
        elif izbrana_kosarica == n-1:
            sosedi = B[n-2:n] + [B[0]]
            minimalni_sosed = min(sosedi)
            indeks = sosedi.index(minimalni_sosed)
            if indeks == 0:
                B[0] += 1
            else:
                B[izbrana_kosarica-2] +=1
            x += 1
        else:
            minimalni_sosed = min(B[izbrana_kosarica -1 : izbrana_kosarica +2])
            indeks = B[izbrana_kosarica - 1 : izbrana_kosarica + 2].index(minimalni_s
            B[izbrana_kosarica -1 + indeks] += 1
            x += 1
    return max(B)
```

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, zato pričakujeva, da bova lahko maksimalno število žog, v odvisnosti od omenjenih parametrov, napovedala z verjetnostjo blizu 1.