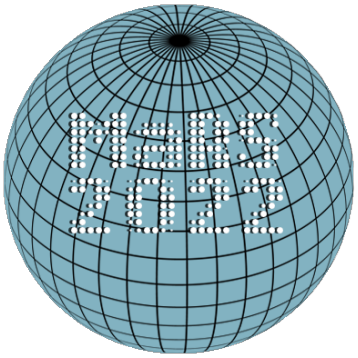
Pošiljanje paketov po d-dimenzionalni kocki

Juš Kocutar, Matej Knap, Kaja Rajter
Mentor: David Opalič



Povzetek

Vsako oglišče d-dimenzionalne kocke želi poslati paket v neko ciljno oglišče. Predstavimo deterministično algoritem in izpostavimo primere, v katerih je počasen. Z dodajanjem naključnega koraka ga "pohitrimo" in dokažemo, da je nov algoritem z veliko verjetnostjo linearen.

1 Uvod

Imamo graf, po katerem pošiljamo pakete. Ti lahko potujejo po povezavah med vozlišči. Po posamezni povezavi lahko potuje le en paket naenkrat. Iščemo algoritem, ki pošilja pakete na njihovo ciljno mesto, ne da se pri tem naberejo prevelike količine paketov na majhnem številu vozlišč in s tem pride do zamud pri pošiljanju paketov. Želimo, da vsako vozlišče izvaja algoritem neodvisno od ostalih vozlišč, kar pomeni, da ne upošteva števila paketov na ostalih voziščih.

Problem je med drugim pomemben v računalništvu, saj je zaželjeno, da podatki potujejo čim hitreje. Mreže vsako sekundo pošiljajo ogromno število podatkov, zato je poznavanje učinkovitega in hitrega algoritma pomembno za delovanje interneta, pošte in marsikaterega drugega sistema. Ker v praksi težko dosežemo, da so vsa vozlišča med seboj komunicirajo in si učinkovito delijo informacije, poskušamo zagotoviti, da vsako vozlišče deluje samostojno in opravlja algoritem neodvisno od ostalih vozlišč.
Naš članek bo tesno sledil poglavju *Chernoff bounds* iz skripte predmeta *Randomized Algorithms and Probabilistic Methods*, ki ga je leta 2021 Angelika Steger predavala na ETH Zurich.

2 Teoretično ozadje

Začnimo z definicijo nekaj verjetnostnih pojmov. Poleg stvari, ki jih bomo definirali, bomo uporabljali še precej terminologije in dejstev iz splošne teorije verjetnosti. Za kratek prelet priporočamo [**1**]. Naj bo $(\Omega, \mathcal{F}, \mathbb{P})$ verjetnostni prostor. Za naše potrebe je dovolj, da je Ω končen. V tem primeru vzamemo σ-algebro $\mathcal{F} = 2^{|\Omega|}$.

Definicija 1. *Realna* ***slučajna spremenljivka*** *je funkcija* $X : \Omega \to \mathbb{R}$.

Eden ključnih pojmov v verjetnosti je neodvisnost. Spomnimo se, da sta dogodka A in B neodvisna, če velja $\mathbb{P}[A \cap B] = \mathbb{P}[A] \cdot \mathbb{P}[B]$. To posplošimo na slučajne spremenljivke na sledeči način.

Definicija 2. *Naključni spremenljivki* X *in* Y *sta* ***neodvisni****, če velja*

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\mathbb{P}(Y = y)$$

za vse $(x, y) \in \mathbb{R}^2$.
V splošnem so naključne spremenljivke $X_1, X_2, \ldots, X_n$ ***neodvisne****, če velja*

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \mathbb{P}(X_1 = x_1)\mathbb{P}(X_2 = x_2) \ldots \mathbb{P}(X_n = x_n)$$

za vse $(x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$.

Za slučajno spremenljivko X s številsko zalogo vrednosti $\mathcal{S}$ definiramo **pričakovano vrednost** kot

$$\mathbb{E}[X] = \sum_{x \in \mathcal{S}} x \cdot \mathbb{P}[X = x].$$

Če zaloga vrednosti ni številska, potrebujemo nekaj teorije mere za definicijo pričakovane vrednosti, v kar pa se ne bomo poglabljali. Pomebna lastnost pričakovane vrednosti, ki jo bomo uporabljali ves čas, je njena linearnost:

$$\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y].$$

Z nekaj dela se lahko pokaže, da za neodvisni slučajni spremenljivki X in Y velja $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$.

Definicija 3. *Naj bo X slučajna spremenljivka. Njena* ***varianca*** *je*

$$\operatorname{Var}[X] = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2.$$

Varianca nam pove, koliko je povprečna absolutna razlika med realizacijo slučajne spremenljivke X in njeno pričakovano vrednostjo. Tako pričakovana vrednost kot varianca lahko zavzameta vrednost ∞.

Definicija 4. *Naj bo $N \geq 1$, $0 \leq p \leq 1$ in naj bo $\Omega = \{1, \ldots, N\}$. Porazdelitev na Ω podano z utežmi, za $0 \leq k \leq N$,*

$$p_k = \binom{N}{k} p^k (1-p)^{N-k}$$

imenujemo ***binomska porazdelitev*** *s parametri p in N. Označimo jo z Binomial(N, p).*

Definicija 5. *Za $d \in \mathbb{N}$, je* ***d-dimenzionalna kocka*** *konveksna kombinacija 2^d različnih točk v d-dimenzijah, ki imajo koordinate 0 ali 1. Tem 2^d točkam pravimo* ***oglišča****. **Rob** kocke tvori par oglišč, ki se med sabo razlikujeta v natanko eni koordinati.*

3 Uvod v problem

V vsakem oglišču d-dimenzionalne kocke je paket, ki mora priti do ciljnega oglišča. Ciljna oglišča so permutacija začetnih oglišč. Vsako oglišče lahko v vsakem koraku pošlje po vsaki povezavi (robu kocke) največ en paket. Povezave so usmerjene, dve oglišči povezuje par povezav v različnih smereh. Naš cilj je ustvariti hiter algoritem, ki bo poslal vsak paket do njegovega ciljnega oglišča. Za algoritem želimo, da je lokalen v smislu, da se mora vsako oglišče odločiti, kam pošlje kateri paket, brez sprejemanja sporočil drugih oglišč oziroma znanja o številu paketov v njih.

V d-dimenzionalni kocki ima vsako oglišče d koordinat z vrednostjo 0 ali 1. Število oglišč je torej 2^d. Rob d-dimenzionalne kocke tvorita dve oglišči, ki se razlikujeta v natanko eni koordinati, torej je vsako oglišče povezano z natanko d drugimi oglišči. Število vseh robov bo torej število oglišč pomnoženo

s številom povezav vsakega oglišča, oziroma $2^d \cdot d$. Spomnimo se, da imamo dve usmerjeni povezavi med sosednjima ogliščema in je število povezav torej dvakrat večje kot pri običajni d-dimenzionalni kocki.

Poskusimo najti algoritem. Naš algoritem se bo moral odločati o dveh stvareh:

- po kateri poti bo poslal pakete in
- v katerem vrstnem redu bo poslal pakete.

Poglejmo si najprej drugi problem. Če oglišče vsebuje dva ali več paketov, ki morajo biti poslani po istem robu, se mora algoritem odločiti o vrstnem redu, v katerem bo pošiljal pakete. Za to bo zadostoval preprost algoritem: FIFO (First in, first out). Oglišče bo pošiljalo pakete v enakem vrstnem redu, kot so paketi prihajali v to oglišče.

Za prvi problem pa si bomo pogledali algoritem *Bitfixing* (pomeni popravljanje bitov). Algoritem bo primerjal koordinate trenutnega oglišča s koordinatami ciljnega oglišča. Koordinate bo preverjal od leve proti desni in koordinato spremenil, če se razlikuje od koordinate ciljnega oglišča, dokler paket ne pride do ciljnega oglišča. Ta algoritem vedno deluje, ne glede na pozicijo začetnega in ciljnega oglišča.
Primer 1. Denimo, da mora paket v 6-dimenzionalni kocki priti od oglišča $(0,1,0,1,1,0)$ do oglišča $(0,1,1,0,1,1)$. Pot bo izgledala tako:

$$(0,1,0,1,1,0) \longrightarrow (0,1,1,1,1,0) \longrightarrow (0,1,1,0,1,0) \longrightarrow (0,1,1,0,1,1).$$

3.1 Problem deterministič nih algoritmov

Algoritem *Bitfixing* vedno deluje, vendar ne nujno hitro. V najslabših primerih mora algoritem izvesti eksponentno število korakov.

Izrek 1. *V najslabšem primeru bo algoritem Bitfixing potreboval vsaj* $\frac{2^{d/2}}{d}$ *korakov.*

Dokaz. Zaradi preglednosti predpostavimo, da je število dimenzij d sodo. Denimo, da morajo vsa oglišča s koordinatami oblike

$$\left(a_1, a_2, \ldots, a_{\frac{d}{2}}, b_1, b_2, \ldots, b_{\frac{d}{2}}\right)$$

poslati pakete v ciljna oglišča s koordinatami oblike

$$\left(b_1, b_2, \ldots, b_{\frac{d}{2}}, a_1, a_2, \ldots, a_{\frac{d}{2}}\right).$$

Ker *Bitfixing* spreminja koordinate z leve proti desni, bodo v tem primeru vsi paketi šli skozi oglišča s koordinatami oblike

$$\left(b_1, b_2, \ldots, b_{\frac{d}{2}}, b_1, b_2, \ldots, b_{\frac{d}{2}}\right).$$

Takih oglišč je $2^{d/2}$. Vsako oglišče lahko v enem koraku pošlje največ d paketov, skozi največ $2^{d/2}$ oglišč pa mora preiti 2^d paketov. Potrebujemo torej več kot

$$\frac{2^d}{d \cdot 2^{d/2}} = \frac{2^{d/2}}{d}$$

korakov. □

Ideja dokaza je, da so včasih nekatera oglišča "preobremenjena". Sprejmejo preveč paketov in zato algoritem potrebuje veliko korakov. Bilo je dokazano (ne enostavno!), da za kateri koli drugi deterministični algoritem velja isti problem, saj je v najslabšem primeru število korakov eksponentno.

4 Neenakosti

Poglejmo si nekaj neenakosti, ki omejijo, koliko se slučajna spremenljivka razlikuje od svoje pričakovane vrednosti. Med drugim so to neenakost Markova, Chebysheva neenakost in Chernoffova neenakost.

Izrek 2 (Neenakost Markova)**.** *Naj bo X nenegativna slučajna spremenljivka. Potem za vsak $t > 0$ velja*

$$\mathbb{P}[X \geq t] \leq \tfrac{\mathbb{E}[X]}{t}.$$

Dokaz. Naj bo $Y = t\mathbb{1}_{\{X \geq t\}}$. Torej je $X \geq Y$. Dobimo

$$\mathbb{E}[X] \geq \mathbb{E}[Y] = \mathbb{E}\left[t\mathbb{1}_{\{X \geq t\}}\right] = t\mathbb{E}\left[\mathbb{1}_{\{X \geq t\}}\right] = t\mathbb{P}[X \geq t].$$

□

Izrek 3 (Chebysheva neenakost)**.** *Naj bo X slučajna spremenljivka, za katero velja $\mathbb{E}[X] < \infty$. Potem za vsak $t > 0$ velja*

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq t\right] \leq \frac{\mathrm{Var}(X)}{t^2}.$$

Dokaz. Velja $(X - \mathbb{E}[X])^2 \geq 0$. Preoblikujemo in uporabimo Markovo neenakost. Dobimo

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq t] = \mathbb{P}[(X - \mathbb{E}[X])^2 \geq t^2] \leq \frac{\mathbb{E}[(X - \mathbb{E}[X])^2]}{t^2} = \frac{\mathrm{Var}(X)}{t^2}.$$

□

Neenakost Markova in Chebysheva veljata za vse slučajne spremenljivke, zato meja, ki jo določata, ni tako stroga. Če slučajnim spremenljivkam dodamo več strukture, recimo določimo njihovo porazdelitev, lahko izpeljemo močnejše neenakosti.

Izrek 4 (Chernoffova neenakost)**.** *Naj bodo $X_1, X_2, \ldots, X_n$ paroma neodvisne slučajne spremenljivke, kjer je $X_i \sim$ Bernoulli(p_i). Naj bo $X = \sum_i X_i$ in $\mu = \mathbb{E}[X]$. Potem za vsak $\delta > 0$ velja*

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu.$$

Dokaz. Za vsak $t > 0$ je funkcija $f(x) = e^{tx}$ injektivna in naraščajoča, zato velja

$$\mathbb{P}[X \geq (1+\delta)\mu] = \mathbb{P}[e^{tX} \geq e^{t(1+\delta)\mu}].$$

Po neenakosti Markova dobimo

$$\mathbb{P}[e^{tX} \geq e^{t(1+\delta)\mu}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}}.$$

Ker so $X_1, X_2, \ldots, X_n$ paroma neodvisne spremenljivke, velja

$$\mathbb{E}[e^{tX}] = \mathbb{E}[e^{t\sum_i X_i}] = \mathbb{E}\left[\prod_i e^{tX_i}\right] = \prod_i \mathbb{E}[e^{tX_i}].$$

Ker je $X_i \sim$ Bernoulli(p_i) imamo

$$\prod_i \mathbb{E}[e^{tX_i}] = \prod_i (p_i e^t + (1-p_i)e^0) = \prod_i (p_i(e^t - 1) + 1).$$

Vemo, da je $e^x \geq x+1$ za vse $x \in \mathbb{R}$, zato velja

$$\prod_i (p_i(e^t-1)+1) \leq \prod_i e^{p_i(e^t-1)}.$$

Torej velja

$$\mathbb{E}[e^{tX}] \leq \prod_i e^{p_i(e^t-1)}.$$

Mejo za $\mathbb{E}[e^{tX}]$ vstavimo v neenakost Markova in dobimo

$$\mathbb{P}[e^{tX} \geq e^{t(1+\delta)\mu}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{t(1+\delta)\mu}} \leq \frac{\prod_i e^{p_i(e^t-1)}}{e^{t(1+\delta)\mu}} = \frac{e^{\sum_i p_i(e^t-1)})}{e^{t(1+\delta)\mu}} = \frac{e^{\mu(e^t-1)}}{e^{t(1+\delta)\mu}}.$$

Neenakost velja za vsak $t > 0$, zato lahko vstavimo $t = \log(1+\delta)$ in dobimo

$$\mathbb{P}[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu.$$

□

V nadaljevanju bomo uporabljali posledice Chernoffove neenakosti, ki sicer ne določajo tako stroge meje, vendar so enostavnejše za uporabo.

Izrek 5 (Posledice Chernoffove neenakosti)**.** *Pod enakimi predpostavkami kot pri Chernoffovi neenakosti veljajo tudi naslednje neenakosti:*

1. $\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\mu\delta^2/3}$ *za* $0 < \delta \leq 1$,
2. $\mathbb{P}[X \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}$ *za* $0 < \delta \leq 1$,
3. $\mathbb{P}[|X-\mu| \geq \delta\mu] \leq 2e^{-\mu\delta^2/3}$ *za* $0 < \delta \leq 1$,
4. $\mathbb{P}[X \geq t] \leq 2^{-t}$ *za* $t \geq 2e\mu$.

Dokaz. Dokažimo le 4 zadnjo neenakost, saj je ta edina, ki jo bomo uporabili. Vemo, da je $\frac{e^\delta}{(1+\delta)^{1+\delta}} \leq (\frac{e}{1+\delta})^{1+\delta}$. Naredimo substitucijo $t = (1+\delta)\mu$ in preuredimo Chernoffovo neenakost. da dobimo

$$\mathbb{P}[X \geq t] = \mathbb{P}[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu} \leq \left(\frac{e}{1+\delta}\right)^t.$$

Želimo, da je izraz na desni strani zadnje neenakosti manjši ali enak 2^{-t}. To pomeni da potrebujemo

$$\frac{e}{1+\delta} \leq \frac{1}{2},$$

kar pa je ekvivalentno naši predpostavki $t \geq 2e\mu$.

□

5 Naključni algoritem

Predstavili bomo prilagojeni algoritem *RandBitfixing* za razporejanje paketov, ki ni deterministič en, ampak najprej razporedi pakete na naključna mesta v d-dimenzionalni kocki.
RandBitfixing ima dve fazi. Prva faza vedno traja $4d$ korakov, druga pa se začne s korakom $4d + 1$.

> **Faza 1:** Za vsako oglišče V_i naključno (neodvisno in enakomerno) izberemo oglišče $\sigma(i)$ v d-dimenzionalni kocki in pošljemo paket iz V_i v $\sigma(i)$ z metodo *Bitfixing*. Paket tam počaka do vključno koraka $4d$.
>
> **Faza 2:** Iz vsakega oglišča $\sigma(i)$ pošljemo vse paket v njihovo prvotno ciljno oglišče, ponovno z metodo *Bitfixing*.

Algoritem je dobro definiran le v primeru, ko v prvi fazi vsi paketi uspejo priti iz V_i v $\sigma(i)$ v največ $4d$ korakih. Nas bodo zanimale le konfiguracije, ko se to zgodi, zato ne bomo razširjali algoritma na ostale primere. Katerakoli dopolnitev deluje.
Obe fazi algoritma sta simetrični, saj v prvi fazi iz prvotne razporeditve pošljemo pakete v naključno razporeditev oglišč (ni nujno permutacija, več paketov lahko gre v isto oglišče), v drugi pa iz naključne razporeditve pošljemo pakete v ciljno razporeditev, edina razlika je torej, da so vse poti "obrnjene".

Želimo dokazati naslednji glavni izrek. Pravi, da je pričakovani čas trajanja algoritma za večino začetnih razporeditev vsaj linearen v d in posledično veliko hitrejši od eksponentnih slabih primerov pri *Bitfixingu*.

Izrek 6. *Z uporabo RandBitfixinga bodo z verjetnostjo vsaj* $1 - 2 \cdot 2^{-2d}$ *vsi paketi prispeli na cilj v največ* $8d$ *korakih.*

Za lažji zapis vpeljimo nekaj notacije.

Definicija 6. *Za vsako oglišče* V_i *naj bo* ρ_i *zaporedje robov, po katerih pride paket iz začetnega oglišča* V_i *v oglišče* $\sigma(i)$ *z metodo Bitfixing. Paket, ki začne v oglišču* V_i*, označimo z* v_i*.*

Recimo, da je pot iz V_i v $\sigma(i)$ dolžine k (na poti je k robov). Potem je ρ_i urejena k-terica robov

$$\rho_i = (e_1, e_2, \ldots, e_k).$$

Dokazali bomo dve lemi, ki opisujeta lastnosti posameznega koraka algoritma. Natančneje bosta opisovali obnašanje poti.

Lema 1. *Recimo, da imata poti ρ_i in ρ_j skupen rob. Potem velja, da ko se poti enkrat ločita, ostaneta ločeni.*

Dokaz. Naj bo V njuno zadnje skupno oglišče v skupnem delu poti. Zaradi metode *Bitfixing* bo tista koordinata oglišč, v kateri sta šli poti po V, od takrat naprej nespremenjena in različna, za oglišča v eni poti 0 in v drugi 1. Tako ne bo več skupnih oglišč v poteh od tod naprej.

□

Osredotočimo se na paket v_i. Naj bo množica S_i enaka

$$S_i = \{V_j \mid 1 \leq j \leq 2^d,\ i \neq j,\ \rho_i \text{ in } \rho_j \text{ imata skupen rob}\}.$$

Naslednja lema bo povezala število korakov, ko se posamezni paket ne bo premikal, s številom poti, ki sekajo ρ_i.

Lema 2. *Paket v_i bo na svoji poti čakal, torej se ne premikal skozi robove, največ $|S_i|$ potez.*

Dokaz. Definirali bomo funkcijo treh spremenljivk *zamuda*, ki bo vsakemu paketu priredila celo število, ki se bo spreminjalo s časom. Naj bo

$$zamuda(t, m, j) = t - m,$$

če gre paket v_j v koraku t skozi rob e_m v poti ρ_i. Ko paket zapusti pot ρ_i ali pa pride na svojo končno oglišče, ki leži nekje na ρ_i, njegova *zamuda* ostane nespremenjena do konca. Zaradi prejšnje leme vemo, da ne bo več prišel nazaj na pot ρ_i, če jo zapusti. Za oglišča, ki nimajo stika z ρ_i in tistimi, ki ga sčasoma imajo, vendar stika še ni bilo, je $zamuda = -\infty$.
Za paket v_i ima *Zamuda* "praktični pomen", saj ob danem trenutku pove, koliko potez je ta do takrat čakal.
Najprej bomo pokazali, da vedno, ko setextitzamuda paketa v_i spremeni iz l v $l + 1$, obstaja paket, ki zapusti ρ_i in ima $zamudo = l$.
Privzamimo, da se *zamuda* paketa v_i poveča z l na $l + 1$. To pomeni, da je paket v_i čakal v nekem oglišču V na poti. Sledi, da se je nek drug paket premaknil iz oglišča V po robu poti ρ_i. Ta paket ima $zamudo = l$, saj se je število korakov povečalo za 1, hkrati pa je šel v naslednje oglišče na poti, zato se *zamuda* ne spremeni. Tako velja, da nekoč v procesu obstaja še en paket poleg v_i na poti z *zamudo* l. Zato obstaja zadnji paket v času z *zamudo* l, ki ali ostane na poti ali jo zapusti. Če ta paket zapusti pot ali ima na njej končno oglišče, smo končali. Če ostane na poti, se mu *zamuda* poveča za 1 in iz tistega oglišča gre nov paket z *zamudo* l, kar je protislovje.

Zato velja, da za vsako povečanje *zamude* paketa v_i iz l na $l+1$, obstaja nekoč v času paket na poti ρ_i, ki ali zapusti pot in se po prejšnji lemi nikoli ne vrne ali pa zaključi svojo pot na ρ_i in je njegova *zamuda* $= l$ v obeh primerih.
Število vseh paketov, ki imajo stik s potjo ρ_i, je $|S_i|$, zato je to tudi največje možno število enolično določenih oglišč z določeno končno *zamudo*. Zato velja, da se *zamuda* paketa V_i poveča za 1 največ $|S_i|$-krat, kar smo želeli pokazati.

□

Lema 3. *Za vsak rob e naj slučajna spremenljivka $T(e)$ označuje število vseh poti skozi rob e v prvi fazi algoritma. Potem velja $\mathbb{E}[T(e)] = \frac{1}{2}$.*

Dokaz. Zaradi simetrije velja $\mathbb{E}[T(e)] = \mathbb{E}[T(e')]$ za vsaka dva robova e in e' v kocki.
Izračunajmo $\sum_e \mathbb{E}[T(e)]$ na dva različna načina. Naj bo R število vseh robov in G število vseh oglišč. Spomnimo se, da je $G = 2^d$ in $R = d \cdot 2^d$.
Če se osredotočino na robove in upoštevamo prejšnjo simetrijo, velja

$$\sum_e \mathbb{E}[T(e)] = R \cdot \mathbb{E}[T(e)].$$

Če se osredotočimo na oglišča, velja

$$\sum_e \mathbb{E}[T(e)] = G \cdot \mathbb{E}[\text{dolžina poti}],$$

ker pošljemo paket iz vsakega oglišča in zato vsakemu oglišču pripada ena pot.
Izračunajmo pričakovano vrednost $\mathbb{E}[\text{dolžina poti}]$. Ker je za vsako oglišče V_i oglišče $\sigma(i)$ izbrano naključno, imamo za vsako koordinato verjetnost $\frac{1}{2}$, da se koordinati V_i in $\sigma(i)$ razlikujeta. Vsaka sprememba koordinate z metodo *Bitfixing* šteje za en korak v poti, zato velja

$$\mathbb{E}[\text{dolžina poti}] = \sum_{\text{vse koordinate}} \mathbb{P}[\text{različni koordinati}] \cdot 1 = \frac{1}{2} \cdot d = \frac{d}{2}.$$

Velja

$$d \cdot 2^d \cdot \mathbb{E}[T(e)] = R \cdot \mathbb{E}[T(e)] = G \cdot \mathbb{E}[\text{dolžina poti}] = 2^d \cdot \frac{d}{2},$$

torej je $\mathbb{E}[T(e)] = \frac{1}{2}$.

□

Sedaj si izberimo in fiksirajmo oglišči V_i in $\sigma(i)$, posledično je fiksirana tudi pot ρ_i. Za vsa ostala oglišča V_j, kjer $j \neq i$, so oglišča $\sigma(j)$ še vedno izbrana naključno.

Izrek 7. *Izberimo in fiksirajmo oglišči V_i in $\sigma(i)$ s potjo ρ_i med njima. Potem velja*

$$\mathbb{P}[|S_i| \geq 3d] \leq 2^{-3d}.$$

Dokaz. Za vse $1 \leq j \leq 2^d$, $i \neq j$, definirajmo slučajno spremenljivko H_j s predpisom

$$H_j = \begin{cases} 1 \ ; \ \rho_i \text{ in } \rho_j \text{ se sekata} \\ 0 \ ; \ \text{sicer} \end{cases} .$$

Spremenljivke H_j so Bernoullijeve in neodvisne, ker so bila oglišča $\sigma(j)$ izbrana paroma neodvisno. Velja tudi $\sum_j H_j = |S_i|$, saj vsaki poti ρ_j pripada natanko eno oglišče, iz katerega je poslan paket na ρ_j. Naj bo

$$\mu = \mathbb{E}\left[\sum_j H_j\right] = \sum_j \mathbb{E}[H_j].$$

Za slučajno spremenljivko $\sum_j H_j$ želimo uporabiti neenakost

$$\mathbb{P}[X \geq t] \leq 2^{-t},$$

ki velja za $t \geq 2e\mu$. To lahko storimo če je X vsota neodvisnih Bernoullijevih naključnih spremenljivk, kar v našem primeru tudi je. Zato želimo preveriti še neenakost

$$3d \geq 2e\mu.$$

Definirajmo slučajno spremnljivko

$$T^i(e) = \begin{cases} T(e) - 1; \text{ če gre pot } \rho_i \text{ skozi rob } e \\ T(e); \text{ sicer} \end{cases} .$$

Velja

$$\sum_j H_j \leq \sum_{l=1}^{k} T^i(e_l),$$

saj na levi strani neenačbe preštejemo, koliko je vseh poti, ki vstopijo v ρ_i, na desni pa število vseh robov ostalih poti v ρ_i, torej najmanj 1 za vsako pot, ki vstopi. Velja

$$\mu = \sum_j \mathbb{E}[H_j] \leq \mathbb{E}\left[\sum_{l=1}^{k} T^i(e_l)\right] \leq k \cdot \frac{1}{2} \leq \frac{d}{2}.$$

V drugi neenakosti smo upoštevali, da velja $\mathbb{E}[T^i(e)] \leq \mathbb{E}(T(e)) = \frac{1}{2}$, v zadnji neenakosti pa $k \leq d$, saj je najdaljša možna dolžina poti enaka d. Velja torej $\mu \leq \frac{d}{2}$, iz česar sledi $3d \geq ed \geq 2e\frac{1}{2}d \geq 2e\mu$. Torej lahko uporabimo četrto posledico Chernoffove neenakosti in dobimo

$$\mathbb{P}\left[\sum_j H_j \geq 3d\right] = \mathbb{P}[|S_i| \geq 3d] \leq 2^{-3d}.$$

□

Zaključimo z dokazom izreka 1.

Dokaz. Z uporabo leme 2 in izreka 7 sledi, da je verjetnost, da ima paket v_i na koncu zamudo več kot $3d$, največ 2^{-3d}.
Ker je celotna pot lahko dolga največ d robov, sledi, da je tudi verjetnost, da paket prispe na cilj v več kot $4d$ korakih (več kot $3d$ za čakanje in največ d za potovanje), največ 2^{-3d}.
Predpostavili smo, da sta oglišče $\sigma(i)$ in posledično pot ρ_i fiksirana, zato zgornja meja za verjetnost velja za vsako izbiro slednjih. To pomeni, da velja tudi za naključno izbiro $\sigma(i)$. Slednje je manj očitno kot izgleda, vendar se da hitro dokazati z uporabo pogojnih pričakovanih vrednosti. Zato lahko predpostavimo, da je verjetnost, da je čas potovanja vsaj $4d$ za naključno izbran v_i, največ 2^{-3d}. Naj bo W_i dogodek, da paket v_i potuje več kot $4d$ korakov. Zanima nas, kolikšna je verjetnost, da obstaja vsaj en paket, ki potuje več kot $4d$ korakov. Torej je verjetnost, da je resničen vsaj en izmed njih, oziroma verjetnost unije vseh dogodkov W_i, enaka

$$\mathbb{P}\left[\bigcup_i W_i\right] \leq \sum_i \mathbb{P}[W_i] \leq 2^d \cdot \mathbb{P}[W_i] \leq 2^d \cdot 2^{-3d} = 2^{-2d},$$

ker je vseh paketov ravno 2^d.
Ker sta oba koraka simetrična, je verjetnost, da vsaj en paket pride na cilj v več kot $8d$ korakih, največ $2 \cdot 2^{-2d}$, kjer smo ponovno uporabili neenakost za unijo dogodkov. Za zaključek moramo vzeti komplement tega dogodka. Drugače povedano, verjetnost, da noben paket ne pride na cilj v več kot $8d$ korakih, je največ $1 - 2 \cdot 2^{-2d}$.

□

6 Zaključek

Predstavili smo problem pošiljanja paketov po d-dimenzionalni kocki in preprost deterministični algoritem za pošiljanje *Bitfixing*. Skonstruirali smo primere ciljnih stanj paketkov, ki povzročijo, da je metoda *Bitfixing* neučinkovita oziroma potrebuje relativno veliko korakov.
Da bi težavo odpravili in algoritem naredili hitrejši v večini ciljnih primerov smo dodali na videz nepotrebno spremembo. Pakete smo v prvi fazi poslali v naključno izbrano oglišče na kocki, v drugi pa iz naključnega v ciljno.
Z uporabo Chernoffove nenakosti smo izračunali, da je z modificiranim algoritmom verjetnost vsaj $1 - 2 \cdot 2^{-2d}$, da bodo vsi paketi prispeli v ciljno oglišče v največ $8d$ korakih. Tako smo ugotovili, da bo naključni algoritem v večini primerov zelo hiter. Čas *RandBitfixing* bo z veliko verjetnostjo linearen v dimneziji d, medtem ko obstaja razred slabih primerov za katere *Bitfixing* potrebuje eksponentno dolgo.
Ali ima smisel primerjati čas potreben za slabe primere deterministčnega algoritma s povprečnim časom naključnega algoritma? V praksi mnogokrat nimamo nadzora nad inputom, ki ga prejemamo. Ta ima neko distribucijo, ki se lahko zgodi, da ima večino mase na slabih primerih. Vse, kar naredimo z našim algoritmom, je, da uniformiziramo distribucijo inputa in si s tem zagotovimo povprečno hitro delovanje algoritma ne glede na začetno distribucijo inputa.

Literatura

[1] J. R. Norris, *Probability*, [ogled 06.09.2022], dostopno na `http://www.statslab.cam.ac.uk/~james/Lectures/p.pdf`