Univerza v Ljubljani
Fakulteta za matematiko in fiziko

Finančni praktikum

Pokrivanje kvadrata s kvadratki z zaporednimi stranicami

Leon Bahovec, Pavla Novak

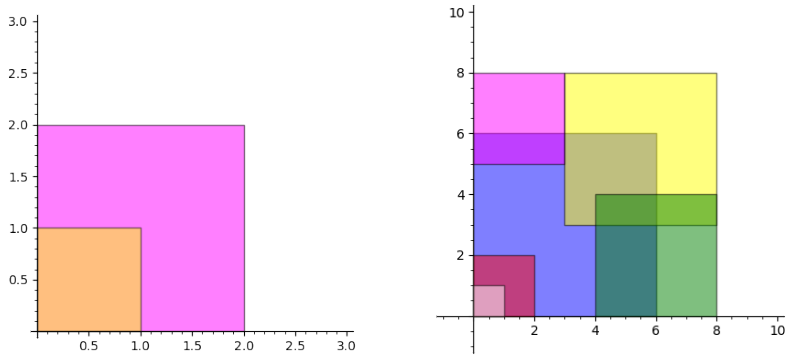
Mentorja: prof. dr. Sergia Cabello Justo, doc. dr. Janoš Vidali

Ljubljana, 2022

Kazalo

1 Priprava

Navodilo naloge je, da imamo za vsako število $i = 1, \dots n$ natanko en kvadratek s stranico dolžine i. Želimo najti največji kvadrat, ki ga lahko pokrijemo z omenjenimi kvadratki. Seveda se lahko prekrivajo, drugače je rešitev veliko manj oziroma za nekatere n-je jih sploh ni. Najmanjši možni n, če se kvadratki ne smejo prekrivati, je 21. Odkril ga je Duijvestijn[1] in obstaja natanko en način, kako zložiti 21 kvadratkov. A vrnimo se na naš problem. Kvadratkov ne smemo rotirati, lahko pa jih premikamo. Problem sva razširila še na večlratno pokrivanje, torej želimo najti največji kvadrat, ki ga lahko vsaj r-krat pokrijemo s kvadratki $i = 1, \dots n$. Oglejmo si primer pokrivanja.



(a) $n = 2$ (b) $n = 6$

Slika a) predstavlja največje možno pokrivanje za $i = 1, 2$ (ne edino možno), b) pa sva domnevala, da bo največje možno pokrivanje za $i = 1, \dots 6$. Vidimo, da sva za $n = 6$ dobila netrivialno rešitev, kvadrat s stranico dolžine 8. A v prihodnje se bo izkazalo, da ta rešitev ni res optimalna. Nalogo sva rešila s celoštevilskim linearnim programom, ki je v nadaljevanju tudi natančneje opisan. Reševala sva na platformi CoCalc in programirala v jeziku SageMath. Oglejmo si CLP. Za $\forall i, u, v$ definiramo spremenljivko $w_{i,u,v}$, ki nam pove, če ima kvadrat s stranico i levi spodnji kot v kvadratku (u, v). $w \in \{0, 1\}$ in pa $\sum_u \sum_v w_{i,u,v} = 1$ (vsak kvadratek je samo en.) Te $w_{i,u,v}$ obstajajo za

$$\forall i = 1, \dots n, u = 1, \dots \sqrt{\frac{n(n+1)(2n+1)}{6}}, v = 1, \dots \sqrt{\frac{n(n+1)(2n+1)}{6}}.$$

Koren zgoraj je največja možna velikost u-ja in v-ja oziroma vsota prvih n kvadratov pod korenom (da dobimo stranico kvadrata). Nato definiramo še $y_{j,k}$, ki nam pove, če je kvadratek (j, k) na mreži pokrit vsaj enkrat. Zavzema vrednosti $\{0, 1\}$. Da bo to res, dodamo naslednjo omejitev $\forall j, k$ na mreži:

$$y_{j,k} \leq \sum_{i=1}^{n} \sum_{u=\max\{1,j-i+1\}}^{j} \sum_{v=\max\{1,k-i+1\}}^{k} w_{i,u,v}$$

[1]Vir: https://mathworld.wolfram.com/PerfectSquareDissection.html

Definiramo še z_l, ki nam pove, če je kvadrat $(0,\dots l)\times(0,\dots l)$ pokrit. Veljati mora naslednja zveza:

$$2lz_l \le z_{l-1} + y_{l,l} + \sum_{m=1}^{l-1}(y_{l,m}+y_{m,l}).$$

Pri tem upoštevamo še robni pogoj $z_0 = 1$. Zdaj definiramo še našo ciljo funkcijo:

$$\max \sum_l z_l.$$

To je osnovni problem, potem pa razširimo še na maksimalni kvadrat, ki je pokrit r-krat. To preprosto zahtevamo, če malce spremenimo prvo omejitev $\forall j,k$:

$$ry_{j,k} \le \sum_{i=1}^{n} \sum_{u=\max\{1,j-i+1\}}^{j} \sum_{v=\max\{1,k-i+1\}}^{k} w_{i,u,v}$$

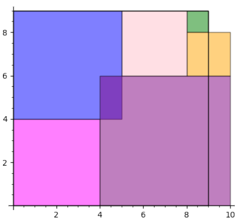
in s tem dobimo vsaj r-kratno pokrivanje. Tu lahko tudi zmanjšamo omejitve, in sicer

$$\forall i = 1,\dots n, u = 1,\dots \frac{\sqrt{\frac{n(n+1)(2n+1)}{6}}}{r}, v = 1,\dots \frac{\sqrt{\frac{n(n+1)(2n+1)}{6}}}{r}.$$

Linearnemu programu sva poleg vseh omejitev in začetnih pogojev dodala še nekaj vrstic kode, da vrne poleg dolžine stranice najdaljšega kvadrataše seznam urejenih trojčkov ('tuples'), ki nam določajo spodnjo levo koordinato vsakega lvadratka in dolžino stranice.

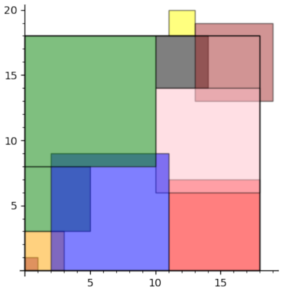
2 Eksperimenti

Najprej sva se hotela prepričati o pravilnosti najnih predivdevanj. Za $n = 2$ je rešitev seveda največja možna, za $n = 6$ pa reštve nisva pravilno uganila. Program je vrnil največji možni kvadrat, ki je v resnici velikosti 9 in ne 8. Oglejmo si še rešitev.

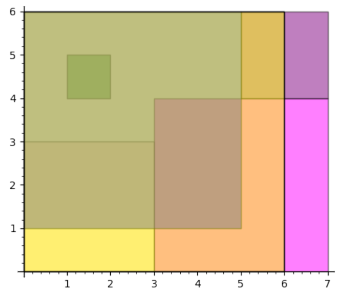


Tu je rešitev oblike `[(1,9,9),(2, 9, 7),(3, 6, 7),(4,1,1),(5,1,5),(6,5,1)]`, kjer prva številka v urejenem trojčku predstavlja stranico kvadrata, druga in tretja številka pa predstavljata spodnjo levo koordinato. S tem je lega in velikost kvadrata v koordnatnem sistemu natančno določena. Seveda nam program ne vrne vseh možnih rešitev, vidimo, da bi lahko naprimer kvadratek s stranico 2 premaknili za eno mesto bolj levo in bi rešitev še vedno bila optimalna.

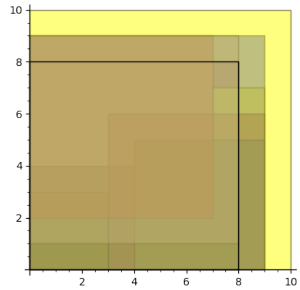
Nato sva poskusila še za $n = 10$. Program je rešitev vrnil v nekaj trenutkih. Izkaže se, da je rešitev kvadrat s stranico velikosti 18. Rešitev, ki jo vrne program, je oblike `[(1,1,1),(2, 12, 19),(3, 1, 1),(4,11,15),(5,1,4),(6,14,14),(7,12,1),(8,11,7),(9,3,1),(10,1,9)]`. Oglejmo si jo še na sliki.



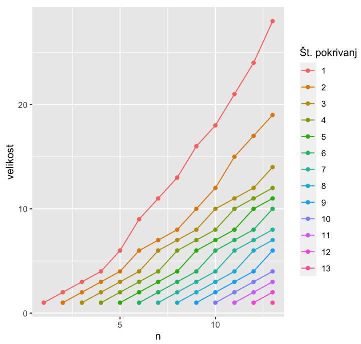
Nato sva poskusila še povečati število minimalnega pokritja, torej kolikokrat mora biti posamezen enotski kvadratek najmanj pokrit. Pričakovala sva, da bo problem postal težji in bo program potreboval več časa, a temu ni bilo tako. Najprej sva pognala program za $n = 6$ in $r = 2$, torej da morajo biti vsi enotski kvadratki v koordinatnem sistemu pokriti 2-krat. Rešitev je `[(1,2,5),(2, 6, 5),(3, 1, 1),(4,4,1),(5,1,2),(6,1,1)]`. Izkaže se, da je največji možni kvadrat z dvakratnim pokrivanjem kvadrat s stranico dolžine 6.



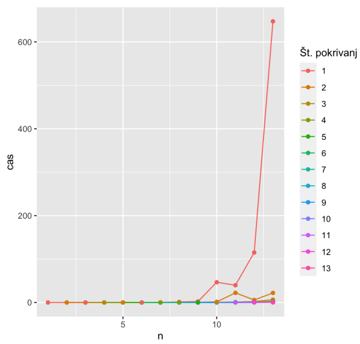
Poskusila sva še s štirikratnim pokrivanjem za $n = 10$, a tudi to programu ni povzročalo težav. Dolžina največjega kvadrata je 8, kar je bistveno manj, kot ko smo zahtevali le enojno pokrivanje, kar je seveda po pričakovanjih. Rešitev je bila `[(1,8,8),(2, 8, 6),(3, 1, 1),(4,1,1),(5,5,1),(6,4,1),(7,1,3),(8,1,2),(9,1,1),(10,1,1)]`. Takšna pa je slika.



Za večje n je problem postal že dokaj zahteven, linearni program je za izračun nekaterih n potreboval tudi več ur, kar sva pričakovala glede na število spremenljivk. Težavnost problema narašča eksponentno. Za večje r sva pričakovala, da bo S pomočjo doc. dr. Janoša Vidalija sva pognala za vse $n = 1, ..20$ in za vsak n še vse možne količine prekrivanj, torej $r = 1, ..n$. Tako sva dobila čas, ki ga najin program potrebuje za vsak zagon, in pa rešitev v obliki od prej. Za nekatere n, naprimer $n = 14, r = 1$ in $n = 15, r = 1$, rezultatov nisva mogla dobiti. Rezultate sva potem zbrala v .txt datoteko in jih v R-ju spravila v grafe, da si lahko bolj nazorno predstavljamo spreminjanje velikosti rešitve(graf a)) in zahtevnosti problema v odvisnosti od n in r (graf b)).

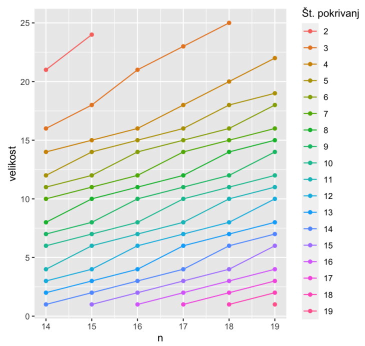


(c) velikost v odvisnosti od n

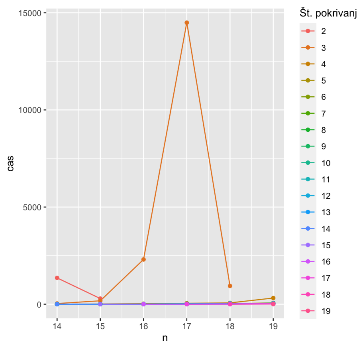


(d) zahtevnost v odvisnosti od n

To so grafi za število kvadratkov do 13. Oba grafa pokažeta obnašanje po pričakovanjih. Z n se maksimalni kvadrat povečuje in s povečevanjem spremenljivke r postaja vedno manjši. Pri grafu b) zelo lepo vidimo, kako problem postaja eksponentno zahtevnejši in program potrebuje več časa.



(e) velikost v odvisnosti od n



(f) zahtevnost v odvisnosti od n

Grafa za $n \geq 14$.

3 Zaključek

Najina naloga je bila napisati linearni program, ki najde rešitev za dan problem pokrivanja kvadrata z manjšimi kvadratki. Ugotovila sva, da je linearni program že za dokaj majhne n počasen. Težavnost se z večanjem števila kvadratkov povečuje, a manjša z večanjem minimalnega pokritja.