Monochromatic crossings in the square

Nino Cajnkar, Nika Pavlič

10.1.2023

1 Opis problema

Naj bo P množica enakomernih naključnih n točk v kvadratu (s stranico dolžine 1). Vsak par točk med seboj povežemo s povezavo in dobimo poln graf z n vozlišči. Za množico kotov $[a, b]$ pobarvamo povezavo rdeče, če je kot, ki ga oklepa povezavo z x-osjo vsebovan v $[a, b]$, sicer pa jo pobarvamo z modro. V nalogi nas zanima število monokromatskih presečišč (to so taka presečišča katerih definirajoči povezave sta obe enake barve) in bikromatskih presečišč (definirajoči povezavi sta različne barve). Zanima nas, kako je to odvisno od velikosti intervala $[a, b]$, predvsem pri katerih intervalih je število bikromatskih vozlišč največje.
Nadalje nas je zanimalo, kako se svari spremenijo, če so točke enakomerno naključno porazdeljene v enotskem krogu.
Kot dodaten problem sva si zastavila vprašanje, kaj pa se zgodi, če v namesto kotov opazujemo dolžine povezav? Tukaj se problem malenkost spremeni, saj iščemo tak interval $[a, b]\,; a, b \in \left[0, \sqrt{2}\right]$ kjer je število monokromatskih vozlišč največje. Enako nas zanima tudi za enotski krog.

2 Podatki

Podatke sva generirala s pomočjo ukaza `rpoint` iz knjižnice `spatstat`, ki generira enakomerne naključne točke, v najinem primeru za kvadrat z dolžino stranice 1. Odločila sva se, da zaradi dolgega časa izvajanja programa pri višjem številu vozlišč obravnavama problem do 50 vozlišč. Tako sva generirala 45 vektorjev točk, sestavljenih iz X in Y koordinate in jih zapisala v datoteko tipa `.json` za lažji dostop v kasnejših delih.
Generirane točke sva nato prebrala iz shranjene `.json` datoteke in s funkcijo

`poln_graf` vsak par točk med seboj povezala in dobila 45 polnih grafov z od 5 vozlišč in do 50 vozlišč.
Sedaj sma morala najti vsa presečišča med povezavami v grafu, pri čemer sva si pomagala s pravilom levega zavoja.

Definicija 1. ***Pravilo levega zavoja:*** *Naj bodo* $P_1 = (x_1, y_1), P_2 = (x_2, y_2), P_3 = (x_3, y_3)$ *in* $P_4 = (x_4, y_4)$ *točke v koordinatnem sistemu in naj bo* p *daljica, ki povezuje* P_1 *in* P_2 *ter* r *daljica, ki povezuje* P_3 *in* P_4*. Definirajmo* ***levi zavoj***

$$Z(P_1, P_2, P_3) = \begin{vmatrix} a_1 & a_{x_1} & a_{y_1} \\ a_1 & a_{x_2} & a_{y_2} \\ a_1 & a_{x_3} & a_{y_3} \end{vmatrix} > 0 \Leftrightarrow (P_1, P_2, P_3) \textit{ je levi zavoj.}$$

Nadalje, premici p *in* r *se sekata natanko tedaj, ko velja*

$$Z(P_1, P_2, P_3) \times Z(P_1, P_2, P_4) \leq 0 \wedge Z(P_3, P_4, P_1) \times Z(P_3, P_4, P_2) \leq 0$$

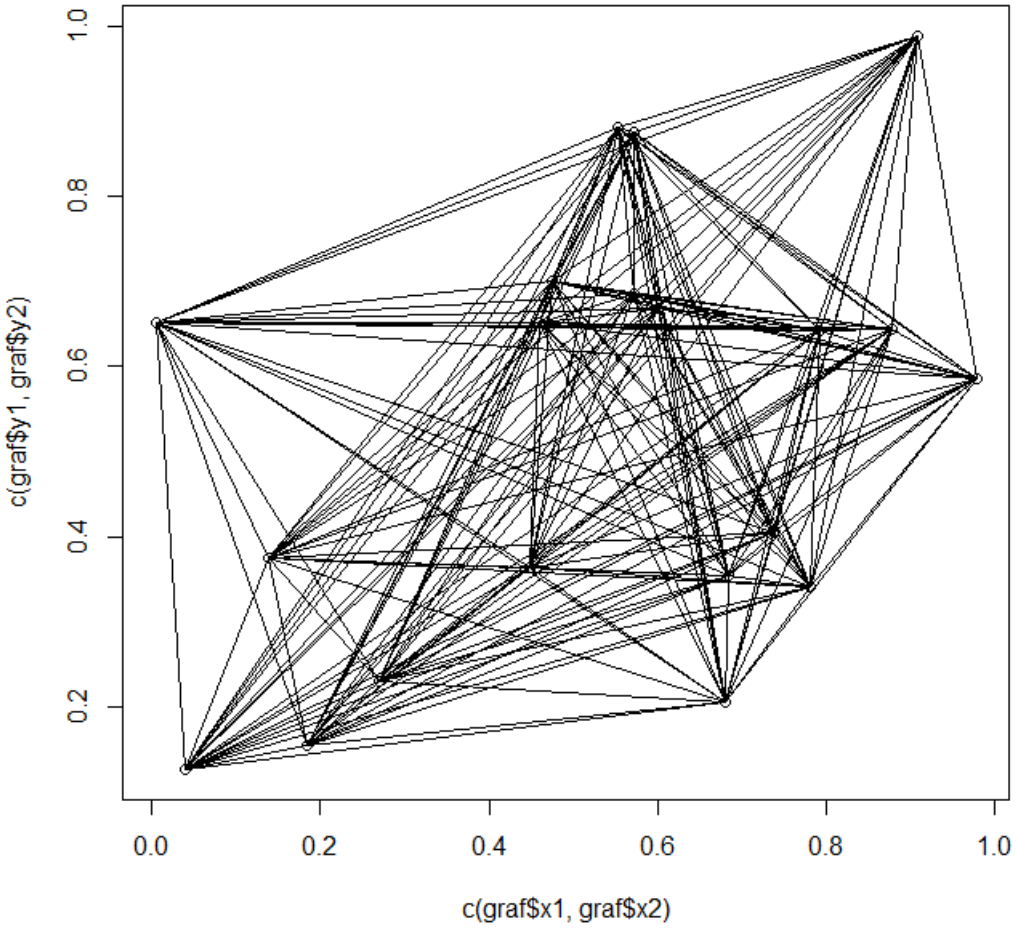
Generiranje točk v krogu se malenkost razlikuje in sicer sma si pomagala s še enim ukazom in knjižnice `statspat` in sicer `runifdisc`. Nadaljni postopek generiranje povezav je enak kot v kvadratu. S funkcijo `presecisca` sva tako poiskala vsa presečišča in jih zapisala v `data frame`, za vsako presečišče sva še izračunala naklon povezav ki ga definirajo ter njihovo dolžino. Postopek ugotavljanja presečišč je enak tudi za točke v enotskem krogu. Podatke o presečiščih za krog in kvadrat sva nato spet prepisala v `.json` datoteko za kasnejši lažji dostop.

3 Reševanje problema

3.1 Barvanje točk v kvadratu glede na interval kotov

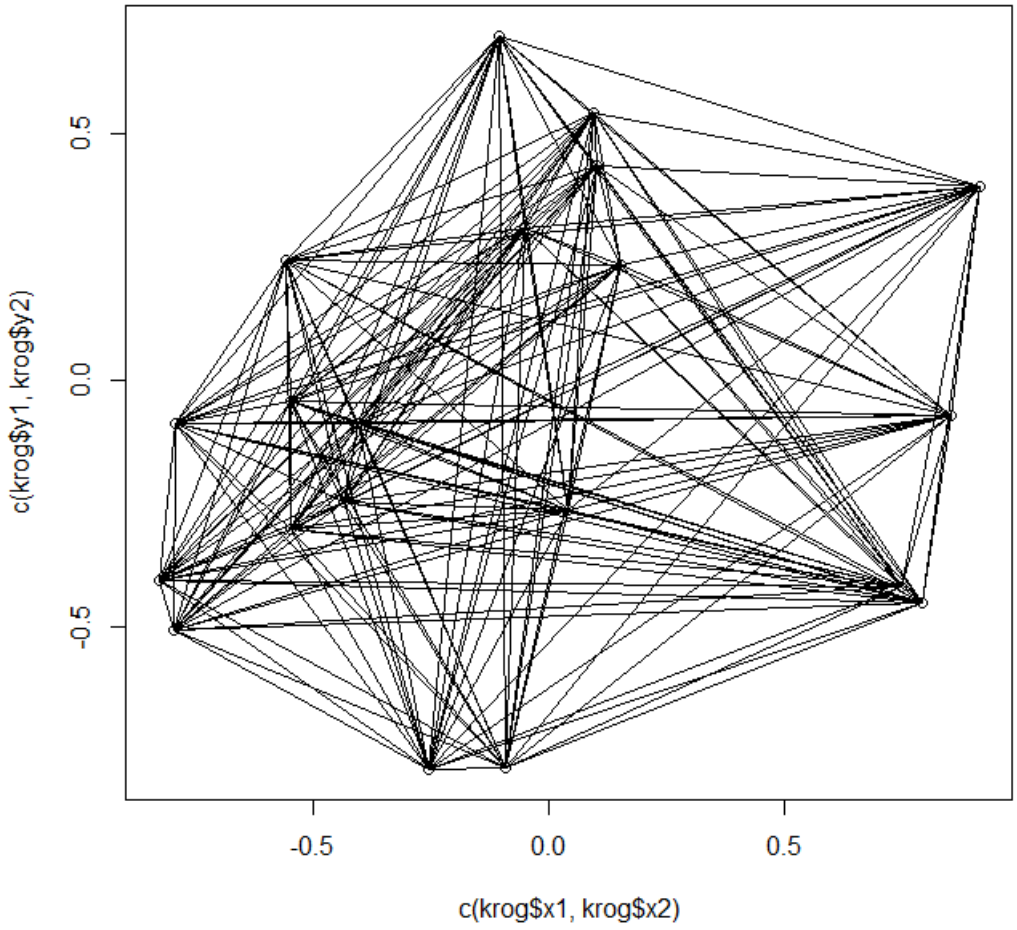
V datotekah tipa `.json` imamo sedaj shranjena presečišča, naklone in dolžine povezav, sedaj pa moramo povezave še pobarvati. In sicer to naredimo s pomočjo funkcije `barvanje`, ki pobarva povezave rdeče, če je njen naklon vsebovan v intervalu $[a, b]\,;\ a, b \in [-\pi/2, \pi/2]$ in modro sicer.
Nadalje definiramo dve funkciji `racunanje_kotov`, ki sprejme podatke o presečiščiš in za intervale $[a, b]$, kjer $a \in [-\pi/2, \pi/2]$ s korakom $\pi/20$ in $b \in [a, \pi/2]$ s korakom $\pi/20$, najprej vsako povezavo obarva rdeče ali modro s funkcijo `barvanje`, nato pa vsakemu presečišču priredi vrednost *mono* ali *bi*, glede na to ali je presečišče monokromatsko ali bikromatsko. Funkcija

Slika 1: Poln graf za enakomerno naključno generiranih 20 točk v kvadratu



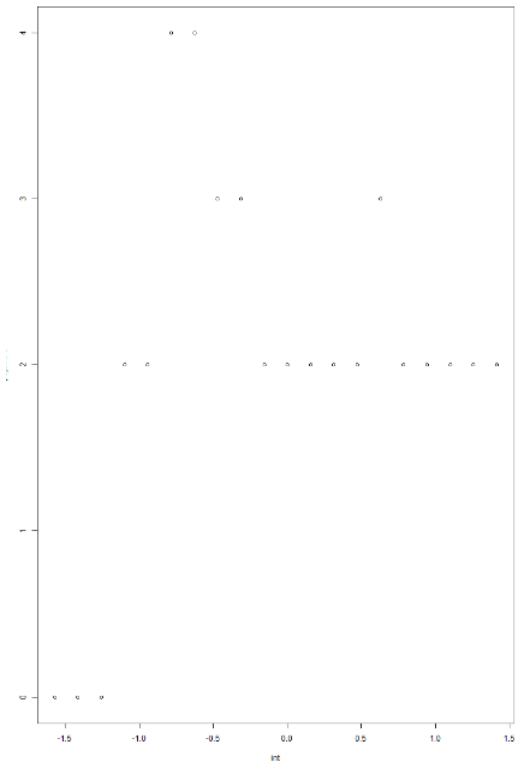
vrne `data frame` intervalov preverjanja ter število monokromatskih presečišč v stolpcu `obarvanost`. Funkcija `racunanje_min_kotov` vrne največji tak interval, v katerem je število monokromatskih presečišč najmanjše.
Ta funkcija je uporabljena za izračun intervalov ter najmanjšega števila monokromatskih presečišč vseh polnih grafov, katere zaporedoma prebere iz datotek `.json`, ter vrne spremenljivko `skupni_min`, v kateri so združeni ti koti ter števila.
Izračunamo še frekvenčni diagram, v katerem zabeležimo začetke intervalov ter število minimumov, ki se znotraj njih pojavijo. To shranimo v spremenljivko `frekvencni`. Za manjše število vozlišč (kar obravnavamo v našem pri-

Slika 2: Poln graf za enakomerno naključno generiranih 20 točk v krogu



meru) opazimo, da največ minimumov najdemo na intervalu $[-\pi/4, -7\pi/20]$ oziroma $[-\pi/4, -\pi/10]$. Lahko bi rezultate še nadalje analizirali, na primer kateri znani porazdelitvi se najbolje prilegajo ter izračunali intervale zaupanja, vendar rezultati ne bi bili zelo natančni zaradi manjšega vzorca.

Slika 3: Frekvenčna analiza intervalov kotov, kjer se pojavijo minimumi monokromatskih presečišč generiranih točk v kvadratu



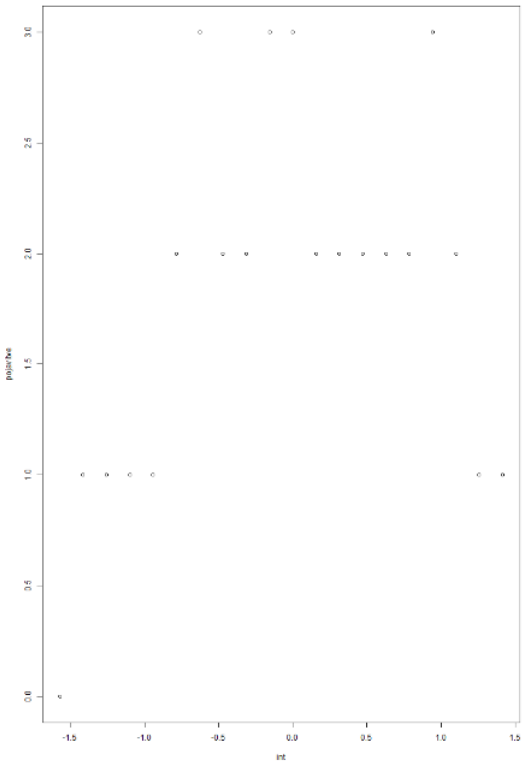
3.2 Barvanje točk v krogu glede na interval kotov

Postopek barvanja generiranih polnih grafov iz točk znotraj enotskega kroga je praktično identičen, vendar dobimo drugačne rezultate. Meje najboljših intervalov so precej premaknjene v pozitivno smer, z največ najdbami minimumov na intervalu $[-\pi/10, \pi/10]$ oziroma $[-\pi/5, \pi/10]$. Rezultati so malenkost manj zanesljivi zaradi osamelca, kar pa lahko pripišemo naključnosti generiranja točk in posledično generiranih grafov, kar bi nadaljna analiza najverjetneje tudi potrdila.

3.3 Barvanje točk v kvadratu glede na intervale dolžin povezav

Tukaj se naloga razlikuje od prejšnje, saj iščemo maksimalno število monokromatskih presečišč.

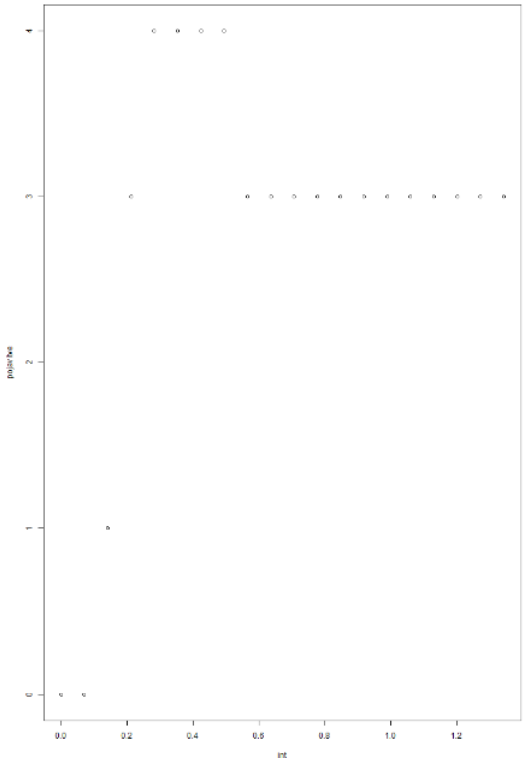
Slika 4: Frekvenčna analiza intervalov kotov, kjer se pojavijo minimumi monokromatskih presečišč generiranih točk v krogu



Tako kot v primeru gledanja kotov imamo v datotekah tipa `.json` shranjena presečišča, naklone in dolžine povezav, sedaj pa moramo povezave še pobarvati. In sicer to naredimo s pomočjo funkcije `barvanje_dol`, ki pobarva povezave rdeče, če je njena dolžina vsebovana v intervalu $[a, b]\,;\ a, b \in \left[0, \sqrt{2}\right]$ in modro sicer.
Nadalje definiramo dve funkciji `racunanje_dolzin`, ki sprejme podatke o presečiščiš in za intervale $[a, b]$, kjer $a \in \left[0, \sqrt{2}\right]$ s korakom $\sqrt{2}/20$ in $b \in \left[a, \sqrt{2}\right]$ s korakom $\sqrt{2}/20$, najprej vsako povezavo obarva rdeče ali modro s funkcijo `barvanje_dol`, nato pa vsakemu presečišču priredi vrednost *mono* ali *bi*, glede na to ali je presečišče monokromatsko ali bikromatsko. Funkcija vrne `data frame` intervalov preverjanja ter število monokromatskih presečišč v stolpcu `obarvanost`. Funkcija `racunanje_max_dolzin` vrne največji tak interval, v katerem je število monokromatskih presečišč največje.
Ta funkcija je uporabljena za izračun intervalov ter največjega števila monokromatskih presečišč vseh polnih grafov, katere zaporedoma prebere iz datotek `.json`, ter vrne spremenljivko `skupni_max_dolzin`, v kateri so združene

te dolžine ter števila.
Izračunamo še frekvenčni diagram, v katerem zabeležimo začetke intervalov ter število maksimumov, ki se znotraj njih pojavijo. To shranimo v spremenljivko `frekvencni_dolzin`. Za manjše število vozlišč (kar obravnavamo v našem primeru) opazimo, da največ maksimumov najdemo na intervalu $\left[\sqrt{2}/5, 2\sqrt{2}/5\right]$. Intervali so precej bolj enakomerno razporejeni kot pri obravnavi intervalov kotov.

Slika 5: Frekvenčna analiza intervalov dolžin, kjer se pojavijo maksimumi monokromatskih presečišč generiranih točk v kvadratu

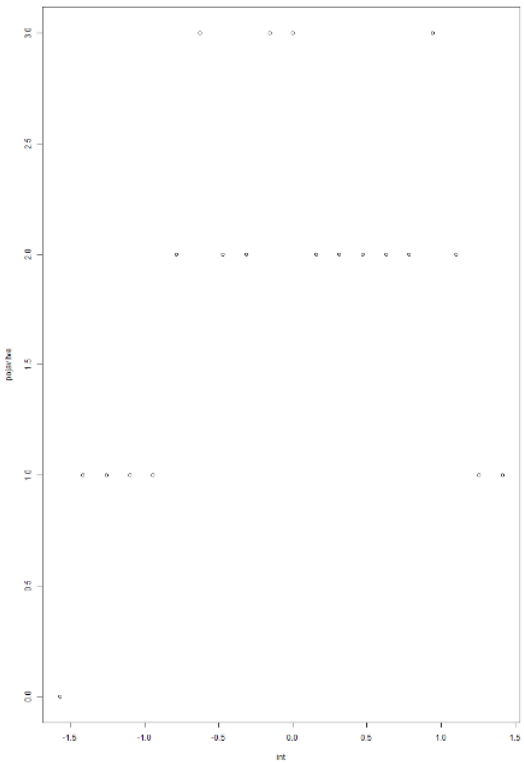


3.4 Barvanje točk v krogu glede na intervale dolžin povezav

Postopek barvanja generiranih polnih grafov iz točk znotraj enotskega kroga je podoben, le da se tukaj možne dolžine povezav razlikujejo, saj je maksimalna možna razdalja enaka premeru enotskega kroga, torej 2. Barvamo torej glede na intervale $[a, b]\,;\ a, b \in [0, 2]$ s korakom 0.1.

Meje najboljših intervalov so precej premaknjene v pozitivno smer, z največ najdbami maksimumov na intervalu $[0.3, 0.7]$ oziroma $[0.3, 1.0]$. Rezultati so tudi tukaj malenkost manj zanesljivi kot pri analizi dolžin v kvadratu, kar pa lahko pripišemo naključnosti generiranja točk in posledično generiranih grafov, kar bi nadaljna analiza najverjetneje tudi potrdila.

Slika 6: Frekvenčna analiza intervalov dolžin, kjer se pojavijo maksimumi monokromatskih presečišč generiranih točk v krogu



4 Zaključek

Za rešitve je bilo generiranih 45 polnih grafov pri vsakem podproblemu, izračunana ter obarvana so bila presečišča ter opravila se je analiza rezultatov. Za natančnejše rezultate bi bilo potrebno generirati ali večje število polnih grafov pri posameznih številih generiranih točk ter polni grafi z večjim številom vozlišč ter povezav. Odkrili smo da se rezultati za točke generirane v kvadratu ter krogu razlikujejo, nadalje bi bilo zanimivo analizirati tudi kako

se stvari nadalje spreminjajo v večkotnikih, na primer da generiramo točke v trikotniku, petkotniku in poljubnem n kotniku. Možnaa naloga bi bila tudi primerjati razlike v rezultatih med kvadratom in krogu očrtanem kvadratu.