VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

INFORMATIKOS KATEDRA

**Grafo šerdis**

**The core of a network**

Kursinis darbas

Atliko: 3 kurso 3 grupės studentas

Miglė Šimavičiūtė (parašas)

Darbo vadovas: dr. Mindaugas Bloznelis (parašas)

Vilnius – 2023

TURINYS

ĮVADAS 1

1. Sąvokos 2

2. Atsitiktinio dvidalio grafo modelis 6

3. Aktorių grafas 6

5. K – šerdies modelis 6

6. Grafo šerdies kitimas kintant rezultatams 10

8. Šaltiniai 11

9. Priedai 13

1 priedas. Pavadinimas 14

2 priedas. Pavadinimas 15

ĮVADAS

Grafų teorija yra matematikos šaka, tirianti ryšius tarp objektų. Grafo struktūra yra svarbi daugeliui praktinių problemų, įskaitant ocialinius tinklus, transporto maršrutus, informacijo perdavimą ir t.t.. Grafo šerdis yra svarbi grafo dalis, nes padeda suprasti grafo savybes bei struktūra.

Šio kursinio tikslas yra ištirti kaip kinta grafo šerdis pagal skirtingius parametrus.

+ dar papildysiu

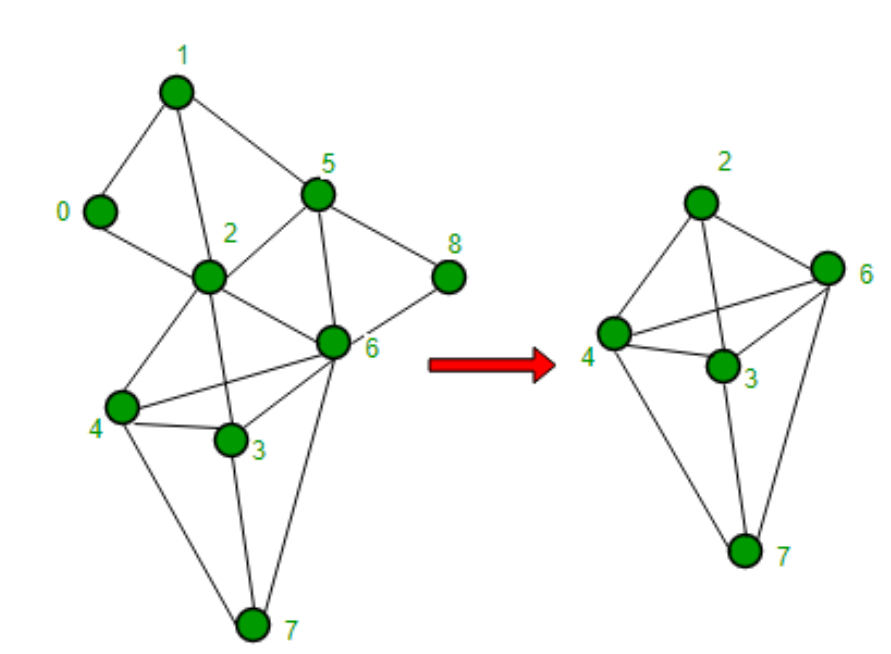
1. SĄVOKOS

Grafas - matematikoje tai abstrakti struktūra, skirta vizualizuoti ryšius tarp skirtingų objektų. Grafas susideda iš viršūnių (V) ir juos jungiančių briaunų (E). Žymėjimas:

Grafo šerdis - tai didžiausia grafo pografis, kuriame kiekviena viršūnė yra susijusi su kita viršūne.

Grafo k-šerdis - tai grafo prografis, kurio kiekviena viršūnė turi k arba daugiau incidenciu briaunu.

Pvz. Taip atrodo grafo (kairėje) k-šerdis, kai k =3 (dešinėje).



Grafo komponentė - tai grafo pografis, kuriame bet kurias dvi viršūnes jungia viena ar daugiau briaunų.

Pvz. Grafas paveiksliuke turės tris komponentes.

A picture containing diagram

Description automatically generated

Dvidalis grafas – tai grafas, kurio viršūnes galime susiskirstiti į dvi nesikertančias grupes taip, kad kiekviena briauna jungtų tik viršūnes iš skirtingų grupių.

+grafo virsuniu aibe

Pvz. Grafą paveiksliuke galime išskirti į dvi viršūnių grupes: (1, 2, 3, 4, 5) ir (A, B, C, E).

A picture containing circle, line

Description automatically generated

Aktorių grafas – tai grafas, kurį gauname iš dvidalio grafo, kurio viena grupę viršūnių laikome aktoriais, o kitą grupę galime laikyti atributais. Aktoriu grafas bus sudarytas iš aktorių viršūnių ir jos tarpusavyje bus susijungusios jei turės bendrą atributą.

2. Dvidalio grafo modelis

Šiame darbe mums reikalingas dvidalis grafas. Tam pasinaudosime modeliu, kuriam sukurti reikalingi du masyvai x ir y, kuriuose bus nurodyti elementų svoriai, taip pat bus reikalingas skaičius α = [0, 1], kuriuo pasinaudosime skaičiuojant tikimybę ar būsimo dvidalio grafo viršūnės bus kaimynės ar ne.

Programai paduodami trys parametrai: du viršūnių masyvai n1 ir n2 su svoriais x ir y bei α.

Algoritmas dvidalio grafo generavimui:

1. Inicializuojamas tuščias sąrašas, kuris bus naudojamas kaip gretimų kaimynių sąrašas.
2. Pradedamas ciklas n1 ilgio kartų, inicializuojamas kintamasis i su pradine reikšme lygia 0.
3. Inicializuojamas kintamasis f su pradine reikšme 0.
4. Pradedamas ciklas n2 ilgio kartų, inicializuojamas kintamasis su pradine reikšme lygia 0.
5. Apskaičiuojamas tikimybės dydis, kuris lygus p = α \* (n1 sąrašo i – asis elementas \* n2 sąrašo f-asis elementas) /
6. Jei atsitiktinis skaičius intervale (0 ; 1) < p, tuomet į gretimų kaimynių sąrašo i – ąjam nariui priskiriama reikšmė j, o j – ąjam nariui priskiriama reikšmė j.
7. Kintamojo f reikšmė padidinama vienetu.
8. Pasibaigus ciklams, sukuriamas tuščias žodynas (angl. dictionary) dvidalisGrafas, kuriame bus saugoma dvidalio grafo informacija.
9. Programa grąžina gretimų kaimynių sąrašą

Pseudokodas:

viršūniųSkaičius = x length + y length

kaimyniųSąrašas = array of empty lists with length viršūniųSkaičius

for i = 0 to x length - 1 do:

f = 0

for j = x length to viršūniųSkaičius - 1 do:

a = random number between 0 and 1

if a < α \* (x[ i ] \* y[ j ]) / sqrt(x length \* y length)) then:

append j to kaimyniųSąrašas [ i ]

append i to kaimyniųSąrašas [ j ]

f = f + 1

dvidalisGrafas = empty dictionary

for i = 0 to viršūniųSkaičius - 1 do:

dvidalisGrafas [ i ] = kaimyniųSąrašas[i]

return dvidalisGrafas

Turėdami sugeneruotą atsitiktinį dvidalį grafą, jį panaudosime aktorių grafo generavimui. Generuodami dvidalį grafą pagal viršūnių svorius bei α galėsime keisti tikimybę ar dvi virūnės bus jungios ar ne, tai padės suprasti kaip kinta šerdis kintant šiems parametrams.

3. Aktorių grafas

Aktorių grafas yra svarbus, nes naudodamiesi jo modeliu galime analizuoti kaip bendri atributai jungia viršūnes. Aktorių grafo gavimui galime pasinaudoti šiuo pseudokodu, kuriam turime paduoti grafo aktorių ilgį ir dvidalį grafą, o programa grąžins aktorių grafą.

Algoritmas aktorių grafo radimui:

1. Iniciliazuojamas tuščias žodynas, kuriame bus saugomi kiekvieno aktoriaus kaimynai.
2. Inicilizuojamas tuščias žodynas, kuria bus saugomas aktorių grafas
3. Pradedamas ciklas, inicilizuojamas kintamasis i su pradine reikšme i = 0.
4. Patikrinama ar aktoriaus kaimyų žodyne yra viršūnė su indeksu 0, jei nėra – pridedama.
5. Kintamojo i reikšmė padidinama vienetu, jei i < aktorių viršūnių skaičių, ciklas tęsiamas, kitu atveju ciklas baigiamas.
6. Pradedamas naujas ciklas, inicializuojamas kintamasis j su pradine reikšme j = 0, kuris bus naudojamas kaip aktorių grafo viršūnė.
7. Sukuriamas žodynas, jei j-asis grafo elementas turi bendrų atributų su kitomis aktorių grafo viršūnėmis – tie kaimynai pridedami į sąrašą, jei ne- tuomet sąrašas tuščias.
8. Kintamojo j reikšmė padidinama vienetu, jei j < aktorių viršūnių skaičių, tuomet ciklas kartojamas, jei ne – ciklas baigiamas.
9. Programa grąžina aktorių grafo gretimų viršūnių sąrašą.

Pseudokodas

kaimynės = {}

for i = 0 to x:

viršūnė, briaunos = list(graph.items())[i]

for kaimynė in briaunos:

if kaimynė not in kaimynės:

kaimynės[kaimynė] = [viršūnė]

else:

kaimynės [kaimynė].append(viršūnė)

aktoriųGrafas = {}

for i = 0 to x:

viršūnė, briaunos = list(graph.items())[i]

vienodiKaimynai = []

for kaimynė in briaunos:

vienodiKaimynai += kaimynai.get(kaimynė, [])

vienodiKaimynai = set(same\_vienodiKaimynai)

vienodiKaimynai.discard(viršūnė)

if vienodiKaimynai:

aktoriųGrafas [viršūnė] = list(vienodiKaiymani)

return aktoriųGrafas

4.K-šerdies radimas aktoriu grafe

Turėdami aktorių grafą, galime jį analizuoti, tam taip pat bus reikalinga k-šerdis, jai rasti naudosime programą, kuri dirba pagal apačioje pateiktą pseudokodą, šerdies radimui programai privaloma pateikti grafą bei skaičių k (>= 0).

Algoritmas k-šerdies radimui aktorių grafe

1. Padaroma duoto grafo kopija, joje bus saugoma grafo šerdis.
2. Apskaičiuojami kiekvienos viršūnės laipsniai ir įrašomi į sąrašą.
3. Pradedamas ciklas, kuris tęsis tol kol bus višrūnių kurias galima pašalinti.
4. Pradedamas naujas ciklas, iniziliacuojamas naujas kintamasis i = 0
5. Jei i – ojo elemento laipsnis mažesnis už k ir didesnis už > 0:
6. Einama per i – ojo elemento kaimynus ir iš kaimynų gretimų kaimynių sąrašo pašaliname i – ąjį elementą.
7. Iš grafo kopijos pašaliname i – ąjį elemntą ir jo kaimynių sąrašą.
8. i – ojo elemento laipsnis sąraše perrašomas į 0.
9. i reikšmė padidinama vienetu.
10. Jei i < grafo viršūnių skaičių, ciklas kartojamas.
11. Programa grąžina grafo kopija su atnaujintais duomenimis, kuriame tik k – šerdis.

Pseudokodas:

+ papildysiu

**5. Grafo šerdies kitimas kintant parametrams**

ŠALTINIAI

PRIEDAI