

Teoria grafurilor si concepte

- Se stabileste cadrul si terminologia pentru algoritmica grafurilor.
- Elementele de baza ale teoriei grafurilor sunt explicate, cu accent pe concepte relevante .
- Se descrie modul în care sunt reprezentate grafurile.
- Se explica diferitele tipuri de grafuri si attributele acestora. Acest lucru este important, deoarece caracteristicile grafului vor determina alegerea algoritmilor grafurilor si ajută la interpretarea corectăa rezultatelor.
- În final, prezentarea generală a tipurilor de algoritmi specifici grafurilor.
 - Terminologia
 - Grafuri: Tipuri si Structuri
 - Random, Small-World, Scale-Free Structures
 - Tipuri de grafuri
 - Grafuri conectate vs. neconectate
 - Grafuri ponderate vs neponderate
 - Grafuri orientate vs. neorientate
 - Grafuri aciclice vs. ciclice. Arbori (Trees)
 - Grafuri Rare (sparse) vs. Dense
 - Grafuri Monopartite, Bipartite, k-Partite
 - Tipuri de Graph Algorithms
 - Pathfinding. Path Types
 - Centrality
 - Community Detection

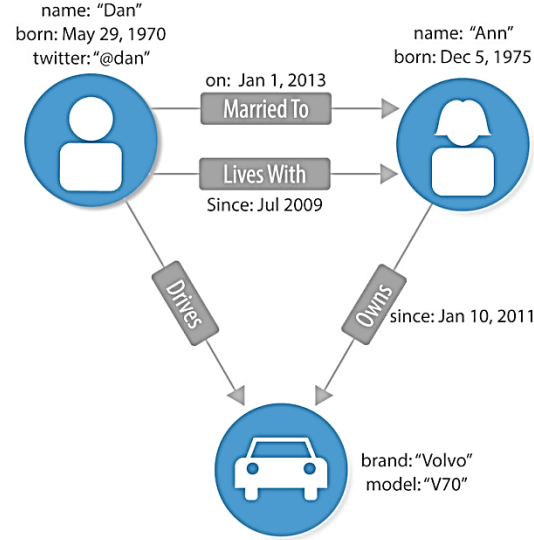
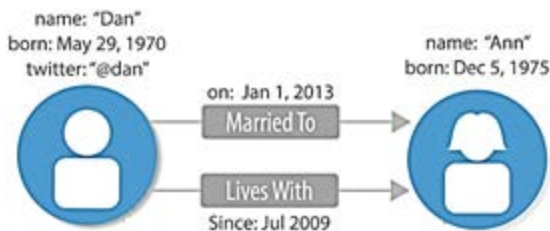
Terminologie

- Graful cu proprietati etichetate este o modalitate uzuală de modelare a datelor folosind grafuri.

- **Eticheta (label)** marcheaza nodul ca parte a unui grup

Figura - Un model de graf cu proprietati etichetat este un mod flexibil si concis de reprezentare a datelor conectate.doua grupuri de noduri: Persoana si Masina (person, car).

- In teoria clasica a grafurilor o eticheta se aplica unui singur nod, acum se foloseste pentru a desemna un grup de noduri.
- **Relatiile sunt clasificate pe baza tipului de relatie.**
- Exemplul include relatia tipuri de DRIVES, OWNS, LIVES_WITH, MARRIED_TO.
- **Proprietatile (Properties)** sunt sinonime cu attributele si pot contine o varietate de tipuri de date, de la numere si siruri la date spatiale si temporale.
- Figura - atribuii de proprietati =perechi nume-valoare, nume proprietate primul si apoi valoarea sa.
- Ex. nod Persoana are nume proprietate "Dan"
 - relatia MARRIED_TO are proprietatea: 1 ianuarie 2013.



Nodes

- Can have *labels* for classification
- Labels have native indexes

Relationships

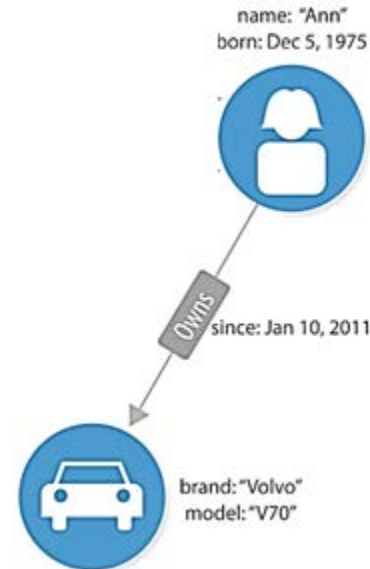
- Relate nodes by *type* and direction

Properties

- *Attributes* of nodes and relationships
- Stored as name-value pairs
- Can have indexes and composite indexes

Un traseu (path) este un grup de noduri si relatiile lor de conectare.

Ex. path: nodurile Ann si Car; DRIVES cu relatiile OWNS

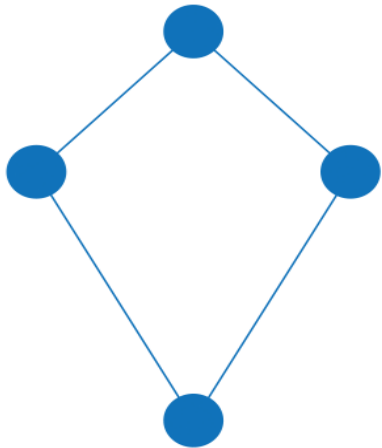


- Un **subgraf** este un graf dintr-un graf mai mare.
- Subgrafurile sunt utile ca filtre, ex. subset cu caracteristici particulare pentru o analiza concentrata.
- Grafurile variaza în functie de tip, forma si dimensiune, si tipul de attribute care pot fi utilizate pentru analiza.

Tipuri si structuri grafuri

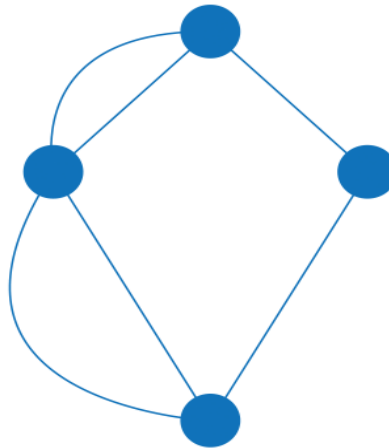
- În teoria clasica a grafurilor, termenul **GRAF** reprezintă un graf simplu (sau strict) unde **nodurile au o singura relatie între ele** (Simple Graph)
- În realitate, majoritatea grafurilor au multe relatii între noduri, chiar relatii de auto-referire.
- Azi, termenul de GRAF este utilizat pentru toate tipurile de grafuri clasice

GRAPH: Simple Graph, MultiGraph, pseudoGraph



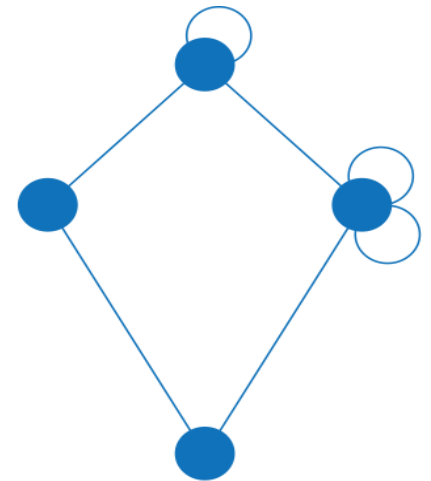
Simple Graph

Node pairs can only have one relationship between them.



Multigraph

Node pairs can have multiple relationships between them.



Graph (also Pseudograph)

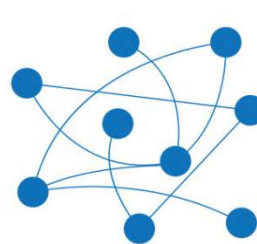
Node pairs can have multiple relationships between them.
Nodes can loop back to themselves.

Random, Small-World, Scale-Free Structures

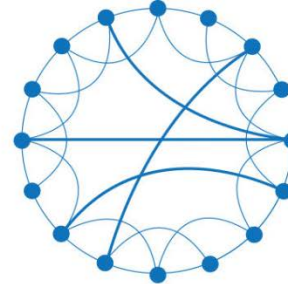
Grafurile au multe forme.

- Retele aleatorii (Random networks)
- Retele la scară redusă (Small-world networks)
- Rețele lipsite de scară (Scale-free networks)
- **Retele aleatorii (Random networks)** Într-o distribuție complet medie a conexiunilor, se formează o rețea aleatorie (random network) fără ierarhii. Acest tip de graf fără formă este "plat", fără modele perceptibile. Toate nodurile au aceeași probabilitate de a fi atașate la orice alt nod.
- **Retele la scară redusă (Small-world networks)** sunt foarte comune în rețelele sociale, afișează localizate conexiuni și un model hub-and-spoke. Cele „6 grade ale lui Kevin” Bacon (actor) ar putea fi cel mai cunoscut exemplu al efectului small-worlds. De-aștră, deși vă asociați mai ales cu un grup mic de prieteni, nu sunteți niciodată mai departe de altcineva - actor celebru sau de cealaltă parte a planetei de 6 legături.
- **O rețea lipsită de scară (Scale-free network)** apare de ex. când există distribuții ale legii energiei electrice și o arhitectură hub-and-spoke este păstrată indiferent de scară, ex. World Wide Web (WWW)
- Aceste tipuri de rețele produc grafuri cu structuri, distribuții distincte și comportamente.
- Pe măsură ce lucrăm cu algoritmi grafurilor, vom ajunge să recunoaștem modele similare în rezultatele noastre.

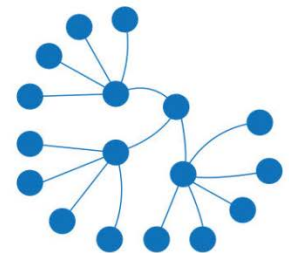
Figura: Trei structuri de rețea cu grafuri și comportamente distincte



Random
Average distributions.
No structure or hierarchical patterns.



Small-World
High local clustering and short
average path lengths.
Hub-and-spoke architecture.



Scale-Free
Hub-and-spoke architecture preserved at
multiple scales.
High power law distribution.

Tipuri de grafuri

Tabelul rezuma caracteristicile/ atributele comune ale grafurilor.

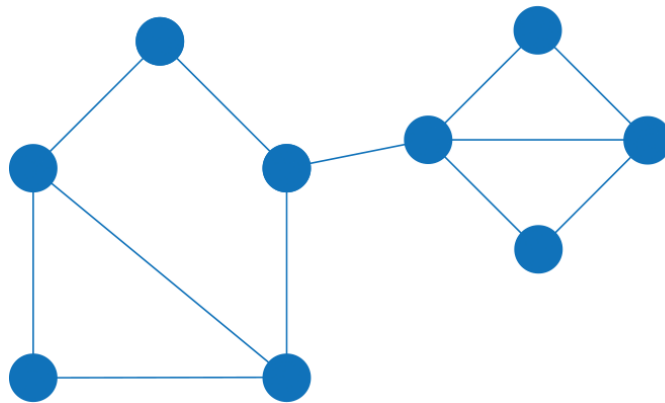
Atribut	Factor cheie	Consideratii
Conectat / neconectat	Se cauta daca este cale (path) intre 2 noduri din graf, fara sa se tina cont de distanta	Noduri/grupuri de noduri izolate pot cauza comportamente neasteptate ale grafului, cum ar fi blocare sau oprirea unui proces prin deconectare componente.
Cost / fara cost Ponderat / neponderat (weighted/unweighted)	Daca exista valori dintr-un domeniu specific in relatiile dintre noduri sau noduri	Multi algoritmi includ optiunea de cost (weight) si exista diferente majore cand costul este ignorat
Graf orientat/neorientat	Daca relatia dintre noduri defineste explicit nod de start sau nod final.	Aceasta adauga continut. In unii algoritmi trebuie specificat explicit graf orientat/neorientat.
Cyclic/ Acyclic	Daca calea (path) incepe si se termina in acelasi nod	Grafurile cicle sunt comune; atentie la stocarea starii la traversare; ciclurile pot preveni terminarea parcurgerii grafului. Grafurile aciclice (sau cele extinse, spanning trees) sunt la baza multor algoritmi.
Spars/dens	Relatii legate de ratia nodurilor	Grafuri extrem de dense pot determina rezultate divergente. Modelarea datelor poate ajuta.
Monpartit/bipartit/k-partit	In functie de nr. de conexiuni: un tip sau mai multe	Folosit: creare de relatii pentru analiza si proiectarea grafuri utile.

Connected Graphs versus Disconnected Graphs

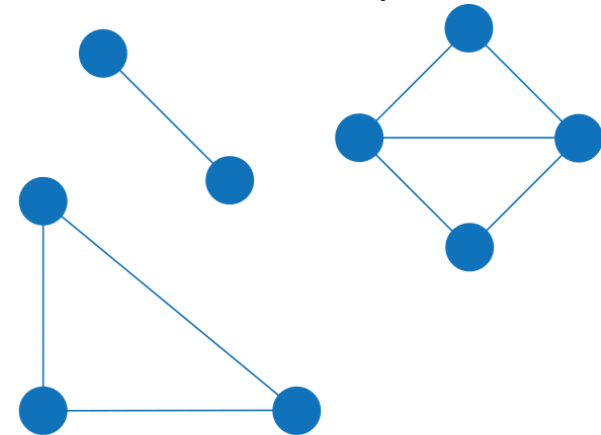
Grafuri conectate vs. neconectate

- **Graf conectat (Connected graph)** = dacă există un traseu între toate nodurile.
- **Graf neconectat (Disconnected graph)** = dacă există insule în graf
- Dacă nodurile din insule sunt conectate, ele sunt numite **componente** (sau, uneori, **cluster**).
- Unii algoritmi au dificultăți cu grafurile neconectate și pot produce rezultate eronate.
- Dacă obținem rezultate neașteptate, primul pas este verificarea structurii grafului.

Figura. Dacă avem insule în graf, este un graf neconectat.
Exemplu de graf neconectat cu 3 componente.



Connected Graph



Disconnected Graph
Includes 3 components.

Unweighted Graphs Versus Weighted Graphs

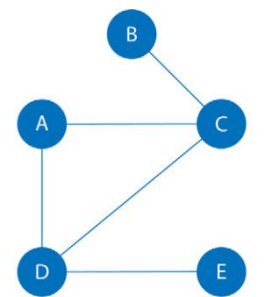
Grafuri neponderate vs. ponderate

- **Grafurile neponderate** nu au valori de greutate atribuite nodurilor sau relațiilor lor.
- Pentru **grafurile ponderate**, aceste valori pot reprezenta o varietate de măsuri, cum ar fi costul, timp, distanța, capacitate sau o prioritizare specifică domeniului.
- Figura-Grafuri neponderate vs. Ponderate. Grafurile ponderate pot conține valori pe relații/noduri.
- Algoritmica grafurilor poate utiliza ponderi pentru procesare ca reprezentare pentru puterea sau valoarea relațiilor.
- Mulți algoritmi calculează valori ce pot fi utilizate ca greutăți în prelucrări ulterioare.
- Unii algoritmi actualizează valorile greutății să continue găsirea de totaluri cumulative, cele mai mici valori sau optime.
- Utilizarea clasică a grafurilor ponderate = algoritmi de pathfinding, utili în aplicații de cartografiere de pe telefoane de ex. calculul cel mai scurt /ieftin /rapid traseu de transport între locații.
- Graf neponderat-calcul cel mai scurt traseu în funcție numărul de relații.

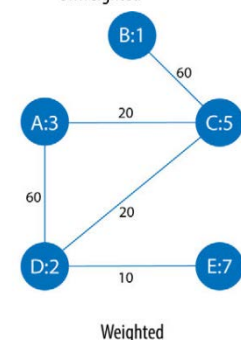
Figura 2 utilizează două metode de calcul al celei mai scurte rute.

- traseele diferă pt. grafurile identice, neponderate și ponderate. (A,D,E 2 hops) vs. (A,C,D,E 50 weight)
- A și E au cel mai scurt traseu= 2 hops, indicând un nod (D) între ele.
- Dar cea mai scurtă cale ponderată de la A la E ne duce de la A la C la D la E. (A,C,D,E 50 weight)
- Dacă greutățile sunt distanță fizică în kilometri.
 - Graf orientat A,C,D,E $AC+CD+DE=20+20+10=50$ weight= 50 km.
 - Graf neorientat- A,D,E (2 hops) echivalent traseu fizic mai lung

$ED+DE=60+10=70$ km.

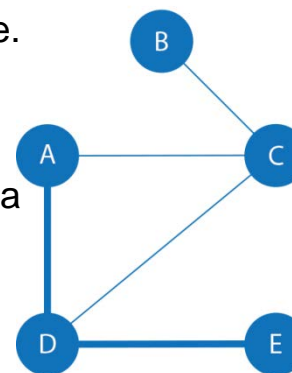


Unweighted

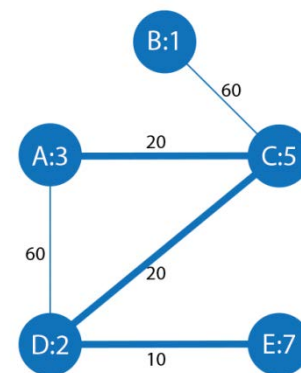


Weighted

What's the Shortest Path Between A and E?



Unweighted
A, D, E = 2 hops



Weighted
A, C, D, E = 50

Undirected Graphs Versus Directed Graphs

Grafuri Neorientate vs. Orientate

Într-un graf neorientat, relațiile sunt considerate bidirectionale (ex. FB friends)

Într-un graf orientat, relațiile au o direcție specifică.

in-links = relațiile care indica un nod

Out-links = relațiile care provin dintr-un nod.

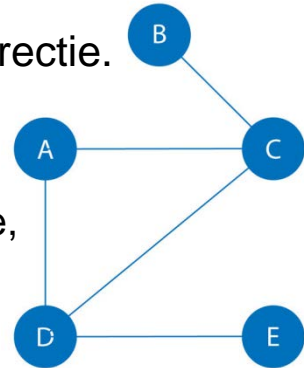
- Direcția, orientarea, adaugă o altă dimensiune informației.
- Relații de același tip, dar în direcții opuse poartă semnificații semantice diferite (different semantic meaning) exprimând o dependență sau indicând un flux.
- Acest lucru poate fi apoi folosit ca indicator al credibilității sau al tăriei grupului.
- Preferințele personale și relațiile sociale se exprimă foarte bine cu factorul direcție.

Ex. figura graf directionat poate reprezenta o rețea de elevi și relațiile "like-uri", se pot calcula (pe baza in/out) A și C sunt cei mai populari elevi.

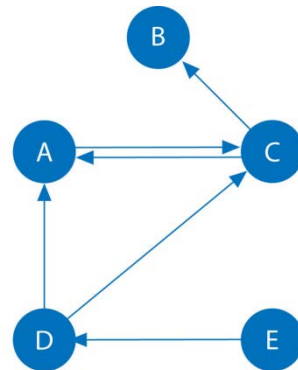
- O multitudine de algoritmi permit calcul doar pe baza conexiunilor de intrare sau ieșire, în ambele direcții sau fără direcție.

Ex. Retelele rutiere ilustrează de ce am putea dori să folosim ambele tipuri de grafuri.

- Autostrăzile dintre orașe sunt adesea circulate în ambele direcții.
- Cu toate acestea, în unele orașe, unele drumuri sunt străzi cu sens unic.
- (Același lucru este valabil și pentru unele fluxuri de informații (information flows!))
- Se obțin **rezultate diferite** ce rulează algoritmi cu graf neorientat sau cu graf orientat.
- În graf neorientat, ex. autostrăzi sau prietenii, s-ar presupune că toate relațiile merg întotdeauna în ambele sensuri.
- Dacă Figura-rețea rutieră orientată, se poate conduce spre A din C și D, dar și doar prin C. Dacă nu ar exista relații de la A la C, ar indica strada blocată. Acest lucru este puțin probabil pentru o rețea rutieră cu sens unic, dar nu pentru un proces sau pagina web.



Undirected



Directed

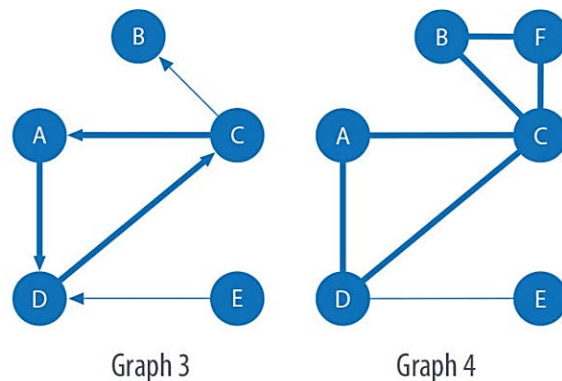
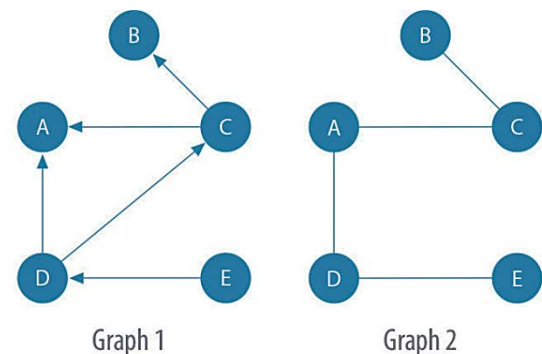
Acyclic Graphs Versus Cyclic Graphs

Grafuri aciclice vs. ciclice

- **Ciclurile (cycles)**= trasee între relații și noduri ce încep și se termină în același nod. (def. teoria grafurilor)
- Graf aciclic (**Acyclic graph**) = graf fără cicluri.
- Graf aciclic direcționat (**Directed acyclic graph (DAG)**) = are totdeauna noduri finale ("**frunze**"=**leaf nodes**)
- Grafurile orientate și neorientate pot avea cicluri, dar cele orientate traseele urmează direcția relației.

Figura

- Grafurile 1 și 2 sunt aciclice, nu au cicluri, deoarece nu există nicio modalitate de a începe și de a termina la același nod fără a repeta o relație.
- Graf 3 - ciclu simplu după A-D-C-A fără nr. noduri repetate.
- Graf 4 - graf ciclic neorientat a devenit interesant prin adăugarea unui nod și a unei relații. Acum există un ciclu închis cu o repetare a nodului (C), după B-F-C-D-A-C-B. Există de fapt mai multe cicluri în graf 4.
- **Pentru a elimina probleme legate de procesare**, uneori trebuie să convertim grafurile ciclice în grafuri aciclice (prin tăierea relațiilor).
- Grafuri aciclice orientate există în mod natural de exemplu în programare, genealogie și istoricul versiunilor..



Trees

Arbori

Arbore (tree) = un graf aciclic care nu este orientat (def. teoria clasică)

- În CS, arborii pot fi și grafuri orientate.

Arbore (tree) = un graf cu oricare doua noduri conectate printr-un singur traseu (def. generală)

- Arborii sunt semnificativi pentru înțelegerea structurii grafurilor si pentru multi algoritmi.
- Au rol-cheie în proiectarea rețelelor (Networks), structurilor de date si optimizari, în cautare pentru a îmbunătăți clasificarea sau ierarhiile organizaționale.

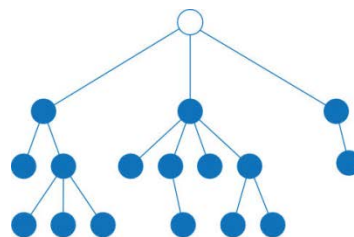
O extindere a arborilor (**spanning tree**)= un subgraf ce include toate nodurile unui graf aciclic mai mare, dar nu toate relațiile. (ex. 3. Arbore extins)

- Arborele de **spanning minim (Minimum Spanning Tree MST)** conecteaza toate nodurile unui graf

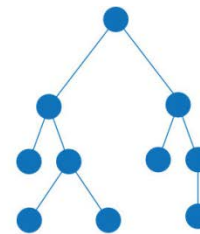
SAU cu cel mai mic numar de salturi intermediare
(either the least number of hops)

(ex. routers, intermediate devices)

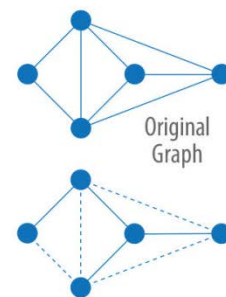
SAU cu traseele cu cele mai mici costuri (least weighted paths)



Rooted Tree
Root node
and no cycles



Binary Tree
Up to 2 child nodes
and no cycles



Spanning Tree
Subgraph of all nodes
but not all relationships
and no cycles

Figura - Arborii cei mai des intalniti in algoritmica grafurilor.

1. Arbore cu radacina fara cicluri - 2. Arbore binar fara cicluri - 3. Arbore extins

Sparse Graphs Versus Dense Graphs

Grafuri Rare vs. Grafuri dense

- **Raritatea** (sparsitatea) grafului = nr. de relatii ale grafului comparativ cu nr. max.posibil de relatii, care ar aparea daca ar exista o relatie între fiecare pereche de noduri.
- **Graf complet (Complete graph, a clique)**= graf în care fiecare nod are o relație cu orice alt nod
(ex. graf complet, clique: daca toti prietenii unei persoane s-ar cunoaste toti intre ei)
- **Densitatea maxima** a unui graf = nr. de relatii posibile într-un graf complet (clique)
- In general orice graf cu o densitate reala care se apropie de densitatea maxima este considerata **densă**.
- Majoritatea grafurilor bazate pe rețele mari reale (ex.FB) tind spre raritate cu o corelatie aproximativ liniara a totalului nodurilor pentru relatii totale.
- Este valabil mai ales în cazul în care apar limitari “fizice” ex. legate de capacitatea tehnologiei: prieteni in rețea.

$$\text{MaxD} = N(N-1) / 2 \quad N \text{ nr. noduri.}$$

- **Densitatea reală**

$$D = 2 R / (N(N-1)) \quad R \text{ nr. relații.}$$

Figura: Ex. de măsuri ale densitatii reale pentru grafuri neorientate. Verificarea densității unui graf va ajuta să evaluați rezultate neașteptate.

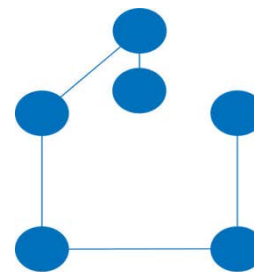
Grafuri prea rare

- este posibil sa nu existe suficiente relatii pentru algoritmi pentru a calcula rezultate utile.
- Unii algoritmi vor returna rezultate fara sens

Grafuri foarte dense

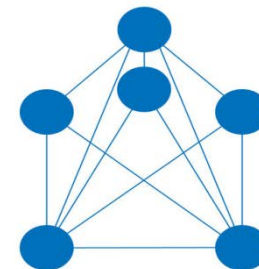
- nu se pot adăuga informatii suplimentare, deoarece sunt excesiv conectate.
- unii algoritmi vor returna rezultate fara sens
- poate adauga complexitate computațională.

Filtrarea subgrafului relevant= soluție, abordare practica



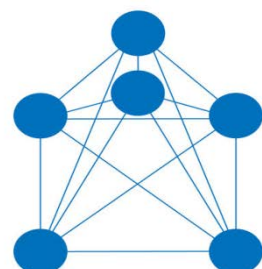
Sparse
Density = 0.3

$$D = \frac{2(5)}{6(6-1)}$$



Dense
Density = 0.8

$$D = \frac{2(12)}{6(6-1)}$$



Complete (Clique)
Density = 1.0

$$D = \frac{2(15)}{6(6-1)}$$

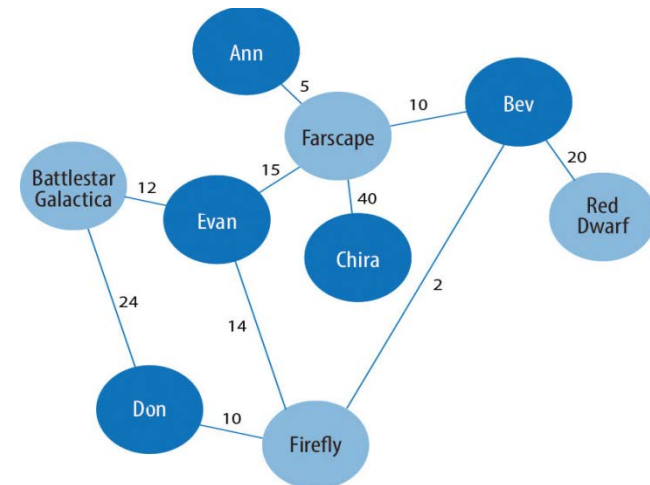
Monopartite, Bipartite, and k-Partite Graphs

Grafuri Monopartite, Bipartite si k-partite

- Majoritatea rețelelor (networks) conțin date cu mai multe tipuri de noduri si relații.
- Algoritmica grafurilor** consideră frecvent un singur tip de nod si un singur tip de relatie.
- Grafuri Monopartite** = grafurile cu un singur tip de nod si un singur tip de relatie.
- Graf bipartit**= graf ale carui noduri pot fi împărțite în doua seturi, astfel încât relațiile conectează doar un nod dintr-un set la un nod dintr-un set diferit.

Ex. figura Graph1

- Include doua seturi de noduri: un set de spectatori (viewers) si un set de TV shows.
- Exista doar relatii între cele doua seturi si nu exista conexiuni intraset.
- Graph 1 include emisiunile TV ce se refera numai la telespectatori, nu si la alte programe TV, iar spectatorii nu sunt, de asemenea, direct legati de alti spectatori.



Graph 1

Viewers and TV Shows

Bipartite Graph

Relationship weights = Number of episodes watched

Figura 11. Grafurile **BIPARTITE** (ex. Graph1 - Viewes and TV shows) sunt proiectate în grafuri **MONOPARTITE** pentru o analiză mult mai specifică.
(ex. Graph2 - Projection of Viewers, Graph 3 -Projection of TV shows)

Din **Graph1** Bipartit - Crearea de 2 Proiectii Monopartite

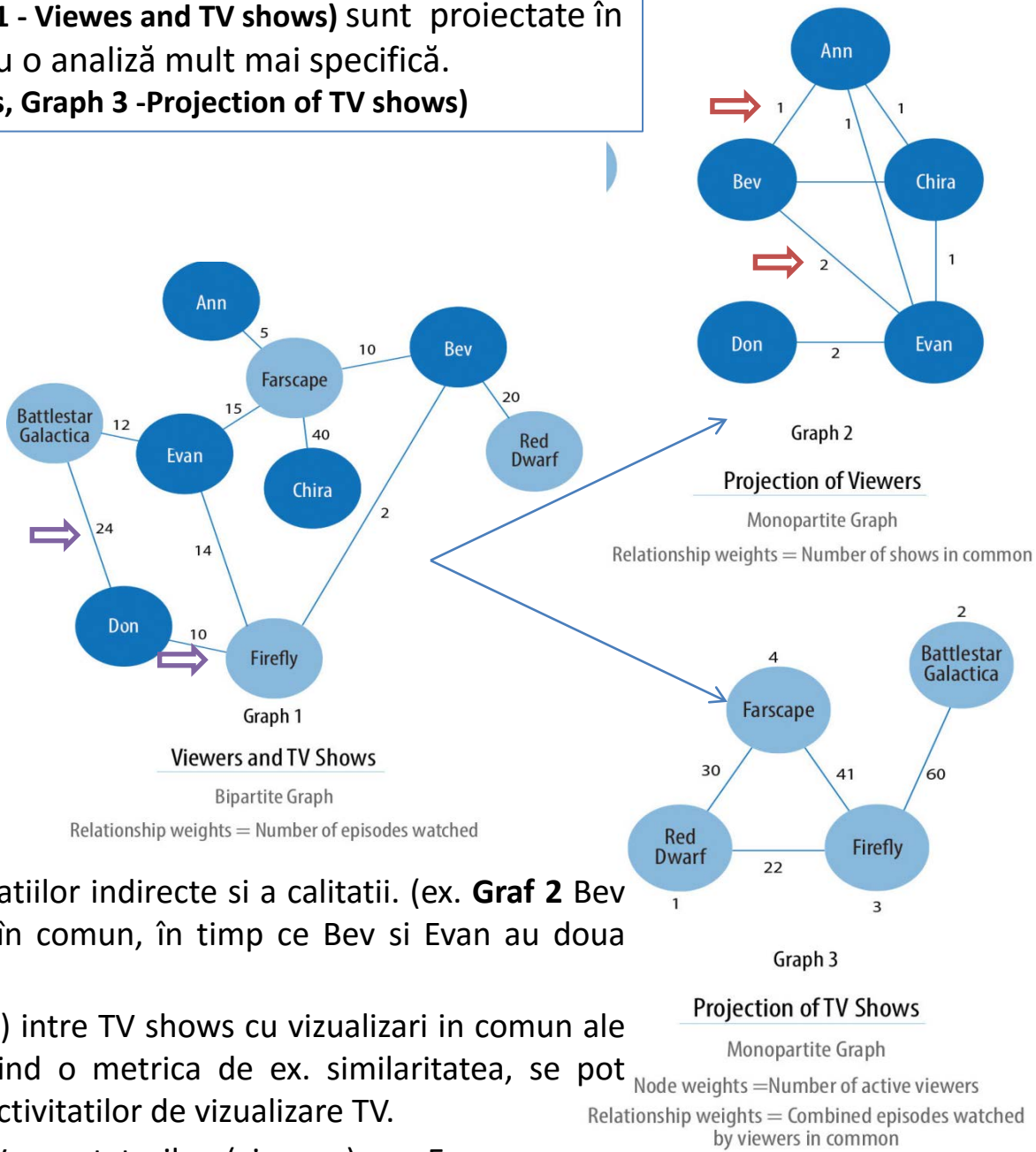
- **Graf 2** al conexiunilor spectatorilor (viewers) bazate pe emisiuni comune (TV shows)
- **Graf 3** al emisiunilor TV bazate pe telespectatori (viewers) în comun.

Se poate aplica **filtru** în funcție de tipul de relatie (**relationship type**) cum ar fi vizionat, evaluat sau recenzat.

Proiectarea grafurilor monopartite cu conexiuni deduse este o parte importanta in analiza grafurilor.

- Aceste proiectii ajuta la descoperirea relatiilor indirecte si a calitatii. (ex. **Graf 2** Bev si Ann au vizionat doar o emisiune TV în comun, în timp ce Bev si Evan au doua emisiuni în comun.) ➡
- **Graph 3** include relatii cu cost (weighted) intre TV shows cu vizualizari in comun ale spectatorilor (viewers). Astfel sau folosind o metrica de ex. similaritatea, se pot utiliza pentru a da un inteles (meaning) activitatilor de vizualizare TV.

Aceasta poate duce la **RECOMANDARI** TV, spectatorilor (viewers) ex. Evan, care a terminat de vzionat (*Battlestar Galactica* si *Firefly*) filme pe o tematica (ex. filme SF) ➡



k-Partite Graphs

Grafurile k-partite

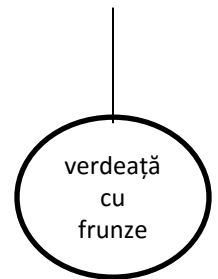
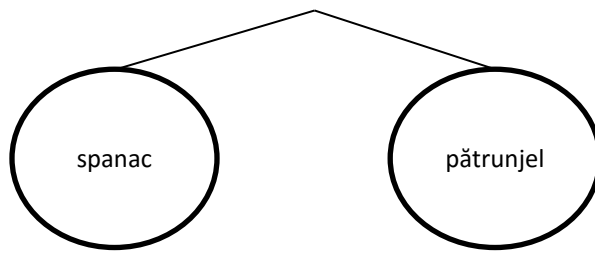
- Grafurile k-partite se refera la numarul de tipuri de noduri pe care le au datele noastre (k).
- De exemplu, daca exista trei tipuri de noduri, am avea un graf tripartit.
- Acestea doar extind conceptul de bipartit si monopartit pentru a ține cont de mai multe tipuri de noduri.
- Multe grafuri din lumea reala, în special grafurile de cunoștințe (**knowledge graphs**), au o valoare mare pentru K, deoarece combină multe diferite concepte si tipuri de informații.

Exemplu de utilizare a unui numar mai mare de noduri

- creare rețete noi prin **maparea unui set de rețete** (ex. Pizza)

la un set de ingrediente a unui compus chimic (ex. set ingrediente sos)

- de aici se vor deduce noi amestecuri ce se vor conecta cu preferințele clienților.
- se pot reduce numarul de tipuri de noduri prin **generalizare**, cum ar fi tratarea mai multor forme ale unui nod, ex. spanac și pătrunjel ca "verdeață cu frunze".



Types of Graph Algorithms

Tipuri de Algoritmi ai Grafurilor

- Se vor prezenta cele trei domenii de analiza care se afla în centrul algoritmilor specifici grafurilor.
- Găsirea traseelor. Tipuri de trasee (Pathfinding. Path Types)
- Centralitatea (Centrality)
- Detectarea Comunităților (Community Detection)

Centrality

Centralitatea

- Centralitatea se refera la înțelegerea nodurilor care sunt mai importante într-o rețea.
- Exista diferite tipuri de algoritmi de centralitate creați pentru a măsura diferite lucruri, cum ar fi capacitatea de a raspândi rapid informații versus crearea unei punți între grupuri distincte.
- Studiul se va concentra asupra modului în care nodurile si relatiile sunt structurate.

Pathfinding

Găsirea traseelor/drumurilor/căilor

- Căile/Traseele sunt fundamentale pentru analiza grafurilor si algoritmi.
- Găsirea celor mai scurte trasee este probabil cea mai frecventă sarcină efectuată cu algoritmi grafurilor si este un precursor pentru mai multe diferite tipuri de analize.
- Cea mai scurta cale este ruta de traversare cu cele mai puține noduri sau cel mai mic cost / greutate / caracteristici-similare (weight), .
- Dacă graful este orientat, atunci este cea mai scurta cale este între două noduri permise de direcțiile relației.

Tipuri de trasee (Path Types)

- **Average Shortest Path** Cel mai scurt traseu mediu este utilizat pentru a lua în considerare eficiența globală și reziliența rețelelor, ex. înțelegerea distanței medii dintre stațiile de metrou.
- Uneori se caută **cel mai lung traseu optimizat** (longest optimized route) pentru situații cum ar fi determinarea stațiilor de metrou care sunt cele mai îndepărtate sau au cele mai multe numărul de opriri între ele, chiar și atunci când este aleasă cea mai bună rută.
- În acest caz, se poate utiliza **diametrul unui graf** (*diameter of a graph*) pentru a găsi cea mai lungă cale cea mai scurtă dintre toate perechile de noduri.

Community Detection

Detectarea comunitatii

- Conectivitatea este un concept de baza al teoriei grafurilor care permite o analiza specific detaliată, cum ar fi gasirea comunitatilor.
- În realitate, majoritatea Rețelelor (Networks) prezinta substructuri (adesea cvasi-fractali) de subgraf mai mult sau mai putin independente.
- Conectivitatea este utilizată pentru a găsi comunități si pentru a cuantifica calitatea grupărilor.
- Evaluarea diferitelor tipuri de comunitati dintr-un graf pot descoperi structuri, cum ar fi hub-uri si ierarhii, si tendinte ale grupurilor de a atrage sau respinge pe altii.
- Aceste tehnici sunt folosite pentru a studia fenomenele emergente, cum ar fi cele care duc la “echo chambers” (efect de ecou al camerei unui grup) si “filter bubble effects” (efect de filtrare a “bulelor” de informații).

Concluzie

In concluzie:

- Grafurile sunt intuitive.
- Grafurile sunt in concordanță cu modul în care gândim si desenam sisteme.
- Principiile lucrului cu grafuri pot fi asimilate rapid odată ce s-a însușit terminologia.