2

Teoria Grafurilor

Teoria grafurilor este studiul grafurilor ca structuri de date care modelează relaţii binare dintre componentele unui sistem. Mai precis, un graf este o pereche ordonată (V, E) în care V este o mulţime de **noduri** iar E o mulţime de **muchii**. Nodurile reprezintă componentele unui sistem, iar muchiile reprezintă legăturile dintre ele.

Această reprezentare simplificată reduce un sistem la o structură abstractă care ne permite să găsim răspunsuri la multe probleme frecvent întâlnite, precum: există un lanţ de conexiuni între două noduri ale grafului? Cum putem găsi un astfel de lanţ de lungime minimă? Câte noduri sunt interconectate între ele? Se poate desena un graf fără a avea intersecţii între muchii? Se poate traversa un graf astfel încât să vizităm fiecare nod o singură dată?

Studiul grafurilor a început în prima jumătate a secolului XVIII cu rezolvarea unor jocuri recreaţionale. A evoluat rapid într-o disciplină majoră a matematicii care a descoperit proprietăţi importante şi utile ale grafurilor şi a găsit soluţii la multe probleme de interes general. Apariţia calculatoarelor a impulsionat studiul algoritmic al grafurilor. Deşi câţiva algoritmi fundamentali au vechime de sute de ani, majoritatea au fost descoperiţi în ultimele decenii. Aceşti algoritmi formează o **tehnologie** fără de care rezolvarea multor probleme importante ar fi practic imposibilă.

Lista următoare ilustrează diversitatea aplicaţiilor bazate pe algoritmi de procesare a grafurilor.

Hărţi

Google Maps şi Via Michelin sunt aplicaţii populare pe care le folosim frecvent ca să aflăm răspunsuri la întrebări precum ,,Care este cel mai scurt traseu de la Timişoara la Bucureşti?" sau ,,Care este cel mai rapid traseu

de mers cu maşina de la Arad la Cluj?" Vom vedea cum aceste probleme se pot rezolva modelând reţeaua rutieră a României (sau Europei) cu grafuri ponderate şi folosind algoritmii lui Dijkstra sau Warshall pentru a găsi drumuri de lungime minimă.

O altă problemă legată de hărţi este cea a colorării regiunilor unei hărţi astfel încât, pentru a le distinge vizual, să atribuim culori diferite regiunilor cu graniţă comună. Harta de mai jos ilustrează că pentru judeţele României sunt suficiente 4 culori.

Aflarea numărului minim de culori pentru care există astfel de colorări este un caz special al problemei de calcul al numărului cromatic al unui graf. Vom prezenta un algoritm de calcul al numărului cromatic, şi vom vedea că hărţile (precum cea a judeţelor României) pot fi reprezentate cu grafuri planare despre care ştim că au numărul cromatic cel mult 4. Deci 4 culori sunt în general suficiente pentru a evita colorarea regiunilor învecinate ale unei hărţi planare cu aceeaşi culoare.

WWW

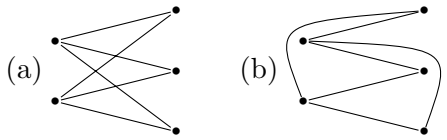
Sistemul informaţional World Wide Web (WWW), numit scurt şi web, este un graf ale cărui noduri sunt pagini web interconectate prin hyperlink-uri. Paginile web sunt documente de tip hipertext care rezidă în diferite locaţii

pe calculatoare server şi pot fi regăsite cu ajutorul unui identificator unic numit URL. Un browser este un program de navigare a grafului WWW care oferă două servicii utile: (1) descarcă pagini web de pe un server şi le afişează pe un terminal client la utilizator, şi (2) permite utilizarea unui mouse pentru a urma hyperlink-uri de trecere de la o pagină la alta. Căutarea informaţiilor stocate în pagini web se face cu motoare de căutare care se bazează pe algoritmi de traversare sistematică a grafului WWW pentru a indexa conţinutul paginilor web şi a reţine aceşti indecşi pe servere din centre de date. În 2016 se estima că centrele de date Google pentru motoare de căutare aveau $2.5 \cdot 10^6$ servere. În septembrie 2020 se estima că sunt peste $5.55 \cdot 10^9$ pagini web indexate pe aceste servere.

WWW şi Internetul sunt sisteme diferite. Internetul este un sistem global de calculatoare între care există legături fizice bazate pe tehnologii de conectare electronică, wireless sau optică, iar comunicarea dintre calculatoare se face protocoalele TCP/IP. Şi Internetul are o structură de graf: nodurile sunt calculatoare iar muchiile sunt legăturile fizice dintre ele.

Circuite electronice

Un circuit electronic este un graf ale cărui noduri sunt resistori, tranzistori şi capacitori conectaţi cu cablaje imprimate (engl. *Printed Circuit Board*). Proiectarea acestor circuite integrate trebuie să găsească rapid răspunsul la întrebarea „Există un scurt-circuit?”, iar în caz afirmativ să răspundă şi la întrebarea „Se poate evita scurt-circuitul printr-o reimprimare a cablajului?” De exemplu, figura (b) este o reimprimare fără scurt-circuit a circuitului din figura (a).



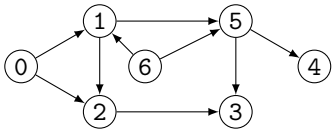
Aceste probleme pot fi rezolvate cu algoritmi de detecţie a grafurilor planare şi de construcţie a unor reprezentări planare (engl. *planar embedding*). Circuitele integrate actuale integrează miliarde de tranzistori, iar proiectarea lor ar fi practic imposibilă fără utilizarea acestor algoritmi.

Probleme de planificare

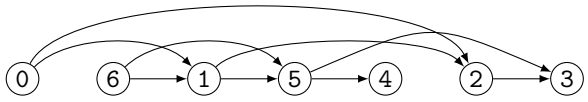
Numeroase procese sunt colecţii de activităţi care trebuie să respecte anumite constrângeri: unele activităţi nu pot începe decât după ce se încheie

alte activităţi. Se doreşte efectuarea acestor activităţi într-un timp cât mai scurt şi să fie respectate constrângerile impuse.

Această problemă de planificare poate fi modelată cu un graf orientat G ale cărui noduri sunt activităţile, iar un arc $x \to y$ constrânge activitatea y să înceapă după încheierea activităţii x. Rezolvarea problemei constă în găsirea unei sortări topologice a lui G, adică a unei enumerări a tuturor nodurilor lui G în o listă L astfel încât, dacă $x \to y$ este un arc în G atunci x apare înaintea lui y în L. De exemplu, graful orientat



are sortarea topologică $[0, 6, 1, 5, 4, 2, 3]$ ilustrată mai jos:



Vom vedea cum calculul unei sortări topologice (dacă există) se poate face rapid cu un algoritm de traversare în adâncime a grafului orientat.

Reţele sociale

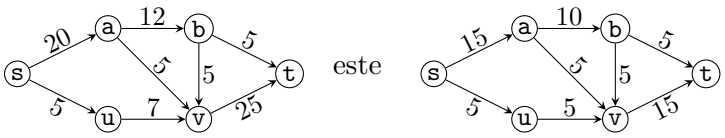
Cu reţelele sociale Facebook sau Twitter stabilim conexiuni directe cu prieteni. O reţea socială este un graf în care nodurile sunt persoane iar muchiile sunt conexiunile stabilite între prieteni sau către persoanele ale căror postări le urmărim. Reţelele sociale sunt analizate cu algoritmi din teoria grafurilor pentru a determina:

- nivelul de conexiune şi densitatea reţelei, prin calculul componentelor conexe, a ordinului, mărimii, şi a gradului minim şi maxim pentru fiecare componentă conexă a reţelei sociale.
- gradul de influenţă al utilizatorilor asupra reţelei, prin calculul unor măsuri de centralitate,
- robusteţea reţelei, prin calculul unor statistici descriptive (de exemplu, gradul de conectivitate),
- preocupările utilizatorilor şi clasificarea lor în grupuri de interes, cu algoritmi de clasificare sau clustering

iar rezultatele analizei pot fi folosite pentru: data mining; marketing; analiza comportamentului utilizatorilor; colectarea de informaţii şi analiza securităţii (supravegherea terorismului şi a crimei organizate); etc.

Fluxuri maxime

Calculul fluxului maxim în reţele de transport apare frecvent în rezolvarea unor probleme de optimizare. Reţelele de transport sunt grafuri orientate ponderate în care există un nod de plecare (nodul *sursă*), un nod de sosire (nodul *destinaţie*), şi nodurile intermediare care sunt puncte de tranzit. Arcele grafului reprezintă legături de transport orientate cu capacităţi de încărcare ce nu pot fi depăşite. Se pune problema găsirii unui flux maxim de transport de la sursă la destinaţie. De exemplu, un flux maxim în reţeaua de transport de mai jos cu sursa `s` şi destinaţia `t`



Reţelele de transport au fost inventate în 1954 ca model simplificat de studiu al fluxului de mărfuri transportate cu trenul în Uniunea Sovietică, dar sunt utile şi pentru îmbunătăţirea condiţiilor de trafic, reconstrucţia imaginilor din proiecţia razelor X în tomografie, planificarea procesoarelor paralele, etc. Vom vedea cum problema fluxului maxim poate fi rezolvată cu algoritmul lui Ford-Fulkerson.

Problema poştaşului chinez

Modelul abstract al acestei probleme este de a găsi un drum sau ciclu care să treacă o singură dată de-a lungul fiecărei muchii a unui grad neorientat. Are aplicaţii în minimizarea costurilor pentru salubrizarea străzilor, colectarea deşeurilor, expedierea poştei, etc. Vom vedea cum putem rezolva această problemă cu algoritmi de detecţie a drumurilor sau ciclurilor euleriene.

Problema comis voiajorului

Dată fiind o listă de oraşe şi distanţele dintre fiecare două oraşe, care este cel mai scurt traseu posibil care vizitează fiecare oraş o singură dată şi se întoarce în oraşul de origine?

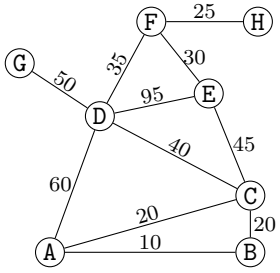
Aceasta este o problemă de optimizare NP-completă[1] pentru care se cunosc algoritmi euristici şi exacţi ce pot găsi soluţiile optime pentru unele cazuri speciale cu zeci de mii de noduri. Pentru cazuri cu milioane de noduri, se cunosc algoritmi care găsesc rapid soluţii suboptimale cu abatere mai mică de 1% (vezi [1]).

Problema comis voiajorului (engl. *travelling salesman problem*) este una dintre cele mai intens studiate probleme de optimizare. A fost formulată matematic în anii 1800 de către matematicianul irlandez Hamilton. Acesta a fost interesat doar de găsirea unui circuit într-un graf neorientat care să treacă o singură dată prin fiecare nod al grafului (un astfel de circuit se numeşte *circuit hamiltonian*). Formularea actuală a problemei a fost dată în anii 1939 de către matematicianul american Flood, care a fost interesat să rezolve o problemă de rutare a autobuzelor şcolare.

Arbori de acoperire cu cost minim

Identificarea arborilor de acoperire cu cost minim apare frecvent în probleme de conectare eficientă a unor puncte de comunicaţie. În aceste situaţii, legăturile posibile dintre punctele de comunicaţie se modelează cu un graf neorientat cu ponderi pozitive ale muchiilor. Soluţia la problemă este un subgraf parţial conex fără cicluri, care (1) conţine toate punctele de comunicaţie şi (2) minimizează suma ponderilor muchiilor sale. Vom vedea că soluţia este întotdeuna un arbore, numit *arbore de acoperire cu cost minim*, care poate fi găsit rapid cu algoritmul lui Kruskal.

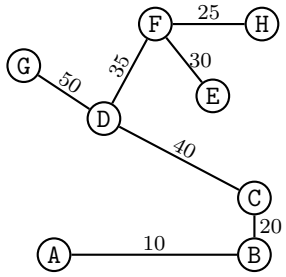
De exemplu, ne putem pune problema conectării cu trenul a localităţilor `A`, `B`, `C`, `D`, `E`, `F`, `G`, `H` între care se pot construi legăturile din graful



Costul investiţiei va fi proporţional su suma lungimilor legăturilor construite. Vrem să investim cât mai puţin şi să existe lanţuri de conexiuni între oricare două localităţi. Cu algoritmul lui Kruskal, putem găsi arborele de acoperire

[1] Acest lucru înseamnă că, în cel mai rău caz, timpul necesar pentru detecţia traseului creşte superpolinomial (dar nu mai mult decât exponenţial) cu numărul de oraşe.

cu cost minim $10 + 20 + 40 + 50 + 35 + 30 + 25 = 210$ ilustrat mai jos:



Cuplaje

Să presupunem că o companie are n muncitori şi tot atâtea locuri de muncă. Fiecare muncitor este calificat pentru unul sau mai multe dintre locurile de muncă disponibile. Se doreşte angajarea unui număr cât mai mare de muncitori pe câte un loc de muncă pentru care este calificat.

Această problemă este un caz special de găsire a unui cuplaj maxim în un graf neorientat. Graful are $2 \cdot n$ noduri care reprezintă muncitorii şi locurile de muncă, şi muchii care conectează fiecare muncitor cu locurile de muncă pentru care este calificat. Un cuplaj este o mulţime de muchii cu proprietatea că oricare două sunt neincidente, adică nu au capăt comun. Un cuplaj maxim are numărul maxim de muchii care cuplează un muncitor cu o muncă pentru care este calificat.

O variaţie a problemei este cea în care ştim şi costurile solicitate de muncitori pentru fiecare serviciu şi vrem să găsim un cuplaj care angajează toţi muncitorii şi minimizează costul total. De exemplu, dacă vrem să angajăm muncitorii Ion, Dan şi Vlad pe posturi de muzician, bucătar şi grădinar, iar costurile percepute de aceştia sunt cele din tabelul de mai jos

Muncitor	Cost muzician	Cost bucătar	Cost grădinar
Ion	108	125	150
Dan	150	135	175
Vlad	122	148	250

atunci avem de găsit un cuplaj maxim cu cost minim în graful ponderat