Grafy i Zastosowania

5: Drzewa Rozpinające

©Marcin Sydow

Spis zagadnień

- drzewa i lasy rozpinające
- cykle fundamentalne i rozcięcia fundamentalne
- własności cykli i rozcięć
- przestrzenie cykli i rozcięć*
- przykład: zastosowanie w sieciach elektrycznych
- minimalne drzewa rozpinające*
- algorytm Kruskala*
- algorytm Prima*
- przykład: aproksymacja dla problemu komiwojażera*

Drzewo rozpinające

drzewo rozpinające spójnego, nieskierowanego grafu prostego $G = (V, E)$ to taki podgraf T tego grafu, który jest drzewem i zawiera wszystkie wierzchołki danego grafu.

Graf niespójny nie posiada drzewa rozpinającego. Jeśli graf G jest niespójny, to graf będący sumą drzew rozpinających jego składowych spójnych (po jednym na składową) nazywamy **lasem rozpinającym**.

przykład

Drzewo rozpinające grafu spójnego można otrzymać kolejno usuwając krawędzie grafu tak aby uzyskać drzewo.

Dla danego grafu spójnego może istnieć wiele drzew rozpinających.

Każde drzewo rozpinające danego grafu ma tyle samo krawędzi (i wierzchołków)

Zliczanie drzew rozpinających grafu spójnego

Tw. (udowodnione przez G.Kirchoffa w 1847)
Liczba różnych drzew rozpinających spójnego grafu etykietowanego wynosi tyle co dopełnienie algebraiczne dowolnego elementu macierzy $M(G) = D(G) - A(G)$, gdzie $D(G)$ jest diagonalną macierzą zawierają na przekątnej stopnie odpowiednich wierzchołków, natomiast $A(G)$ jest macierzą sąsiedztwa grafu G.

przykład

Powyższe twierdzenie ilustruje techniki tzw. *algebraicznej teorii grafów*, gdzie wyniki dla grafów uzyskuje się badając macierze sąsiedztwa i inne algebraiczne struktury reprezentujące grafy.

Liczba cyklomatyczna i rząd rozcięcia

liczba cyklomatyczna: $\gamma(G)$, (lub rząd cykliczności) grafu G to liczba krawędzi dopełnienia dowolnego lasu rozpinającego grafu G.

rząd rozcięcia: $\xi(G)$, grafu G to liczba krawędzi w dowolnym lesie rozpinającym G.

przykład

Zauważmy, że: $\gamma(G) + \xi(G) = |E(G)|$ (liczba krawędzi grafu G)

Cykle a rozcięcia

Twierdzenie:
Jeśli L jest lasem rozpinającym grafu G, to:

1. każdy cykl w G ma wspólną krawędź z dopełnieniem L
2. każde rozcięcie grafu G ma wspólną krawędź z L

Dowód:

1. jeśli cykl nie ma krawędzi wspólnych z dopełnieniem L, to znaczy jest w nim zawarty, co przeczyłoby acykliczności L
2. rozcięcie powoduje rozpad L na dwie składowe A i B. Ponieważ L jest lasem rozpinającym, więc musi zawierać krawędź łączącą pewien wierzchołek z A z pewnym wierzchołkiem z B. Jest to szukana krawędź

Cykle fundamentalne

Niech L oznacza pewien las rozpinający grafu G,

Zauważmy, że dodanie jakiejkolwiek krawędzi z G nie należącej do L utworzy dokładnie jeden cykl.

Taki cykl nazywamy **cyklem fundamentalnym** grafu G związanym z lasem rozpinającym L.

Zbiór cykli fundamentalnych związanych z lasem L to zbiór wszystkich takich cykli.

przykład

Rozcięcia fundamentalne

Niech L oznacza pewien las rozpinający grafu G,

Gdy z lasu L usuniemy dowolną krawędź, to (w odpowiadającej jej składowej spójnej) powstają dwa rozłączne zbiory wierzchołków V_1, V_2.

Zbiór wszystkich krawędzi G takich, że jeden koniec jest w V_1 a drugi w V_2 tworzy rozcięcie, które nazywamy **rozcięciem fundamentalnym** związanym z lasem L.

Zbiór wszystkich takich rozcięć nazywamy **zbiorem rozcięć fundamentalnych** związanych z lasem L.

Uwaga: zbiór rozcięć fundamentalnych niekoniecznie zawiera wszystkie rozcięcia (np. jeśli rozcięcie jest częścią drzewa rozpinającego)

przykład

Własności cykli i rozcięć

Poniżej zakładamy dla uproszczenia, że graf jest spójny (wyniki mogą być zastosowane do każdej składowej spójnej)

Fakty:

- zbiór krawędzi jest rozspajający $\Leftrightarrow$ przecina się z każdym drzewem rozpinającym (ale niekoniecznie minimalny, bo można by wziąć całe E)
- zbiór krawędzi C grafu G zawiera cykl $\Leftrightarrow$ dopełnienie każdego drzewa rozpinającego w G przecina się z C
- cykl i rozcięcie mają zawsze parzystą liczbę wspólnych krawędzi (0 też jest parzysta)

Dalsze własności cykli i rozcięć

Twierdzenie:
T jest drzewem rozpinającym, C jest cyklem fundamentalnym otrzymanym z T przez dodanie krawędzi e. Wtedy C składa się z e i tych krawędzi T, które wyznaczają fundamentalne rozcięcia zawierające e

przykład

Twierdzenie:
Rozcięcie fundamentalne wyznaczone przez odjęcie krawędzi e z drzewa rozpinającego T składa się z e i dokładnie tych krawędzi w dopełnieniu T, których cykle fundamentalne zawierają e.

Przestrzeń krawędzi

$W_e(G)$ - zbiór wszystkich podzbiorów $E(G)$

operacja sumy prostej na elementach $W_e(G)$:
$E_1 \oplus E_2 = (E_1 \setminus E_2) \cup (E_2 \setminus E_1)$
(różnica symetryczna)

przykład

Fakt:
$W_e(G)$ z operacją $\oplus$ jest przestrzenią liniową nad ciałem Z_2.
Bazę stanowi tu zbiór $E(G)$ wszystkich krawędzi grafu G.

przykład

Podprzestrzeń cykli $W_C(G)$

Elementy to zbiór pusty, i zbiory krawędzi wszystkich cykli G i sum cykli krawędziowo rozłącznych. (elementy $W_C(G)$ można nazywać **cyklami uogólnionymi**)

przykład

Twierdzenie:
$W_C(G)$ jest podprzestrzenią liniową przestrzeni $W_E(G)$ (w szczególności, jest zamnkięta na sumę).

Fakt:
graf jest eulerowski $\Leftrightarrow$ jego zbiór krawędzi jest cyklem uogólnionym

przykład

Podprzestrzeń rozcięć $W_S(G)$

Elementy to: zbiór pusty, zbiory krawędzi wszystkich rozcięć i sum krawędziowo rozłącznych rozcięć.

$W_S(G)$ stanowi podprzestrzeń liniową przestrzeni $W_E(G)$. (w szczególności, jest zamnkięta na sumę)

przykład

Baza przestrzeni liniowej (przypomnienie)

Baza przestrzeni liniowej to taki podzbiór elementów przestrzeni liniowej, że:

- generuje całą przestrzeń
- jest liniowo niezależny

przykład

Uwaga: każdy element przestrzeni liniowej jest w dokładnie jeden sposób wyrażalny jako kombinacja liniowa elementów bazy.

Uwaga 2: wymiar przestrzeni liniowej to liczba elementów bazy (każda baza ma tyle samo elementów).

Bazy podprzestrzeni $W_C(G)$ i $W_S(G)$

Twierdzenie:
Zbiór cykli fundamentalnych dowolnego drzewa rozpinającego stanowi *bazę* podprzestrzeni cykli $W_C(G)$.

przykład

Twierdzenie:
Zbiór rozcięć fundamentalnych dowolnego drzewa rozpinającego stanowi *bazę* przestrzeni $W_S(G)$

przykład

Wniosek:
Wymiar przestrzeni $W_C(G)$ wynosi $\gamma(G)$, a wymiar przestrzeni $W_S(G)$ wynosi $\xi(G)$.

Zależności między przestrzeniami cykli i rozcięć

Tw:
Każdy element przestrzeni cykli $W_C(G)$ ma parzystą liczbę krawędzi wspólnych z każdym elementem przestrzeni rozcięć $W_S(G)$ i odwrotnie.

Wniosek:
Przestrzenie $W_C(G)$ i $W_S(G)$ są *ortogonalnymi* podprzestrzeniami przestrzeni krawędzi $W_E(G)$. (tzn. iloczyn skalarny dowolnych par z odpowiednich zbiorów daje zero, ponieważ jest to parzysta liczba jedynek)

Zliczanie cykli i rozcięć

Wnioski:

- w grafie G istnieje dokładnie $2^{\gamma(G)}$ różnych cykli uogólnionych.
- w grafie G istnieje dokładnie $2^{\xi(G)}$ różnych podgrafów, z których każdy jest rozcięciem lub sumą rozcięć krawędziowo rozłącznych.

przykłady

Własności macierzy incydencji

G: nieskierowany graf prosty, wszystkie operacje są nad ciałem Z_2. Niech D oznacza pewien zbiór krawędzi grafu G. Przez I_D oznaczamy zbiór kolumn macierzy incydencji odpowiadających zbiorowi krawędzi D.

Fakty:

- D stanowi cykl uogólniony $\Leftrightarrow$ suma kolumn w I_D wynosi 0
- D reprezentuje graf acykliczny $\Leftrightarrow$ kolumny w I_D są niezależne liniowo
- D reprezentuje podgraw spójny $\Leftrightarrow$ kolumny w I_D rozpinają całą przestrzeń kolumn macierzy incydencji
- D reprezentuje drzewo rozpinające $\Leftrightarrow$ kolumny w I_D stanowią bazę przestrzeni kolumn całej macierzy incydencji

Przykładowe zastosowanie: sieci elektryczne

Dana jest sieć: topologia + opory + przyłożone napięcie
wyznaczyć: natężenia prądu

przykład

prawo Ohma: $U = I \cdot R$
(U - napięcie, I - natężenie, R - oporność)

Dwa prawa Kirchoffa:

1. dla węzłów sieci: suma natężeń w węźle wynosi 0
2. dla oczek sieci: suma napięć w oczku sieci wynosi 0

przykład

Układ równań dla sieci elektrycznej

Można więc potencjalnie ułożyć aż $n + o$ równań (gdzie o to liczba węzłów a o to liczba różnych oczek sieci).

Problemem jest to, że istnieje potencjalnie bardzo dużo cykli w grafie (jak już wiemy jest to dokładnie $2^{\gamma(G)}$)

Większość równań jest redundantna, gdyż potrzebujemy dokładnie tyle równań ile jest krawędzi w grafie.

Zastosowanie cykli fundamentalnych

Ile więc dokładnie równań potrzebujemy?

1. n-1 dla pierwszego prawa (n-te równanie jest redundantne)
2. $\gamma(G)$ dla drugiego prawa

(zauważmy, że faktycznie $(n-1) = \xi(G)$ a więc otrzymamy dokładnie tyle ile trzeba $(\gamma(G) + \xi(G) = |E(G)|)$.

Które równania dla oczek wybrać?

Rozwiązanie: wybrać równania **odpowiadające dowolnemu zbiorowi cykli fundamentalnych tej sieci**

przykład

Problem: Minimalne drzewo rozpinające (MST)

Dany jest nieskierowany graf G z wagami na krawędziach (liczby wymierne).

Znaleźć drzewo rozpinające o minimalnym łącznym koszcie krawędzi (tzw **minimalne drzewo rozpinające**)

przykład

Problem ten ma rozliczne zastosowania. Jest on rozwiązywalny w czasie wielomianowym. Algorytmy znajdowania MST oparte są na własnościach cykli i rozcięć (Kruskal) lub na modyfikacji algorytmu BFS (Prim).

Agorytm Prima (przypomnienie)

Zaczyna od wierzchołka startowego s i stopniowo powiększa drzewo rozpinające. Niech S oznacza zbiór wierzchołków rosnącego drzewa. Poczatkowo $S = \{s\}$. Zauważmy, że zbiór krawędzi o dokładnie jednym końcu w S stanowi zbiór rozspajający. W każdym kroku dodawany jest wierzchołek będący drugim końcem najlżejszej krawędzi z tego zbioru rozspajającego.

Używana jest kolejka priorytetowa, aby efektywnie znaleźć taki wierzchołek (priorytetem jest waga najlżejszej krawędzi łączącej ten wierzchołek ze zbiorem S). Po wybraniu, wszystkie krawędzie wychodzące z nowo-dodanego wierzchołka poddawane są relaksacji.

Algorytm Prima

$w(u,v)$ oznacza wagę krawędzi (u,v), w atrybucie `dist` przechowywana jest najkrótsza aktualnie znana odległość wierzchołka do zbioru S, a `pq` oznacza kolejkę priorytetową. Używamy list sąsiedztwa. Wynikowe drzewo reprezentowane jest w atrybutach `parent`.

```
MSTPrim(V,w,s){
  PriorityQueue pq
  s.dist = 0
  s.parent = null
  pq.insert(s)
  for each u in V\{s}:
    u.dist = INFINITY

  while(!pq.isEmpty()):
    u = pq.deleteMin()
    u.dist = 0
    for each v in u.adjList:
      if (w(u,v) < v.dist):
        v.dist = w(u,v)
        v.parent = u
        if (pq.contains(v)): pq.decreaseKey(v)
        else pq.insert(v)
}
```

Analiza algorytmu Prima

rozmiar danych: n=|V|, m=|E|

dominująca operacja: przypisanie (w inicjalizacji) i porównanie priorytetów (w tym ukryte w kolejce) i odległości

inicjalizacja: $O(n)$
pętla: (n $\times$ delMin()) + (m $\times$ decreaseKey())

Jeśli kolejka zaimplementowana jako kopiec binarny:
pętla: $O(nlog(n)) + O(mlog(n)) =$ **O((n+m)log(n))**

Jeśli używamy kopca Fibonacciego (amortyzowany koszt stały operacji decreaseKey()):
$O(nlog(n) + m)$

Algorytm Kruskala (przypomnienie)

1. początkowo $T = \emptyset$
2. rozpatruj krawędzie w kolejności niemalejących wag i dodawaj do T te, które nie tworzą cyklu z poprzednio dodanymi, pozostałe odrzucaj, do momentu, gdy T nie tworzy drzewa rozpinającego

Główny problem to efektywne sprawdzanie, czy rozpatrywana krawędź nie tworzy cyklu z dotychczasowo dodanymi.

Pomysł polega na używaniu pomocniczej struktury danych typu *union-find*. Ponieważ w każdej iteracji T stanowi las, każda nowa krawędź (u, v), która utworzyłaby cykl ma tę własność, że oba jej końce u i v należą do tego samego drzewa w lesie T.

Algorytm Kruskala

```
kruskalMST(V,E,w){
  T = 0
  UnionFind uf
  foreach edge (u,v) in non-decreasing order of weight:
    if (uf.find(u) != uf.find(v)):
      T = T + (u,v)
      uf.union(uf.find(u),uf.find(v))
  return T
}
```

Istnieje bardzo szybka (drzewowa) implementacja struktury union-find, która zapewnia stały czas operacji `union` i prawie[1] stały amortyzowany czas operacji `find`. Analiza złożoności czasowej tej implementacji nie jest jednak matematycznie łatwa. Przy takiej implementacji złożoność jest $O(mlog(m))$ (i jest zdominowana przez początkowe posortowanie krawędzi po wagach)

[1] jest to pewna funkcja, która bardzo wolno rośnie

Przykład: problem Komiwojażera (TSP)

Dany jest pełny graf $G = (V, E)$ z nieujemnymi wagami $w : E \rightarrow Q^{+}$ na krawędziach. Znaleźć cykl Hamiltona H w G o minimalnym łącznym koszcie krawędzi $w(E(H))$.

przykład

Problem ten jest NP-trudny. Ma on wiele zastosowań praktycznych.

MST jako przybliżenie dla metrycznego TSP

Można w czasie wielomianowym znaleźć *przybliżone* o współczynnik co najwyżej 2 rozwiązanie dla TSP jeśli funkcja wag w spełnia nierówność trójkąta.

Algorytm:

- znaleźć dowolne drzewo rozpinające i zastąpić każdą krawędź parą krawędzi przerciwnych
- znaleźć cykl Eulera w takim grafie
- pominąć (stosując skróty) w tym cyklu wszystkie wierzchołki, które występowałyby wielokrotnie

przykład

(jest to przykład tzw **algorytmu aproksymacyjnego** ze **współczynnikiem aproksymacji** 2. Istnieje też wielomianowy algorytm aproksymacyjny dla tego problemu z lepszym współczynnikiem aproksymacji 3/2 (istnieją jeszcze lepsze).

Podsumowanie

- drzewa i lasy rozpinające
- cykle fundamentalne i rozcięcia fundamentalne
- własności cykli i rozcięć
- przestrzenie cykli i rozcięć*
- przykład: zastosowanie w sieciach elektrycznych
- minimalne drzewa rozpinające*
- algorytm Kruskala*
- algorytm Prima*
- przykład: aproksymacja dla problemu komiwojażera*

Przykładowe ćwiczenia i zadania

- podaj wartości $\gamma(G)$, $\xi(G)$ dla podanego grafu G
- wyznacz zbiór cykli i rozcięć fundamentalnych danego grafu
- mając daną sieć elektryczną z podanym napięciem i opornościami, oblicz natężenia prądów na wszystkich połączeniach (krawędziach) sieci
- wyznacz drzewo rozpinające danego grafu używając algorytmu Prima/Kruskala
- oblicz ile jest cykli i rozcięć uogólnionych w danym grafie
- oblicz ile jest drzew rozpinających w danym grafie etykietowanym
- podaj rozwiązanie przybliżone metrycznego problemu komiwojażera w podanym grafie

Dziękuję za uwagę