Drzewa spinające

- podstawowe definicje
- minimalne drzewa spinające:
  - algorytm Prima
  - algorytm Kruskala

Drzewa spinające

Def. Niech G będzie grafem spójnym o n wierzchołkach. *Drzewo spinające* grafu G to dowolny jego n-wierzchołkowy podgraf będący drzewem. Jeśli G jest niespójny, to powyższy podgraf nazywamy *lasem spinającym*.

Def. *Liczba cyklomatyczna* grafu G to liczba krawędzi, których usunięcie z G prowadzi do utworzenia drzewa (lasu) spinającego i jest oznaczana symbolem $\gamma(G)$.

Przykłady

$\gamma(T) = 0$, T – drzewo,

$\gamma(L) = 0$, L – las,

$\gamma(U) = 1$, U – graf jednocykliczny,

$\gamma(K_n) = n\,(n-1)/2 - n + 1$,

$\gamma(W_n) = n - 1$.

Drzewa spinające

Tw. *Dla dowolnego grafu G o k składowych spójności zachodzi*

$$\gamma(G) = m - n + k.$$

Dowód: Rozważmy najpierw przypadek, gdy G jest spójny. Wiadomo, że n wierzchołkowe drzewo ma $n - 1$ krawędzi. Aby w grafie G znaleźć podgraf będący drzewem spinającym, usuwamy wszystkie krawędzie, z wyjątkiem tych $n - 1$, co oznacza, że musimy usunąć $m - n + 1$ krawędzi.

Załóżmy teraz, że G nie jest spójny. Wówczas $G = G_1 \cup \ldots \cup G_k$, dla pewnego $k > 1$. Możemy założyć, że każdy podgraf G_i jest spójny. Udowodniliśmy powyżej, że $\gamma(G_i) = m_i - n_i + 1$, dla każdego $i = 1, \ldots ,k$, gdzie $n_i = |V(G_i)|$, $m_i = |E(G_i)|$. Dodając równania stronami otrzymujemy:

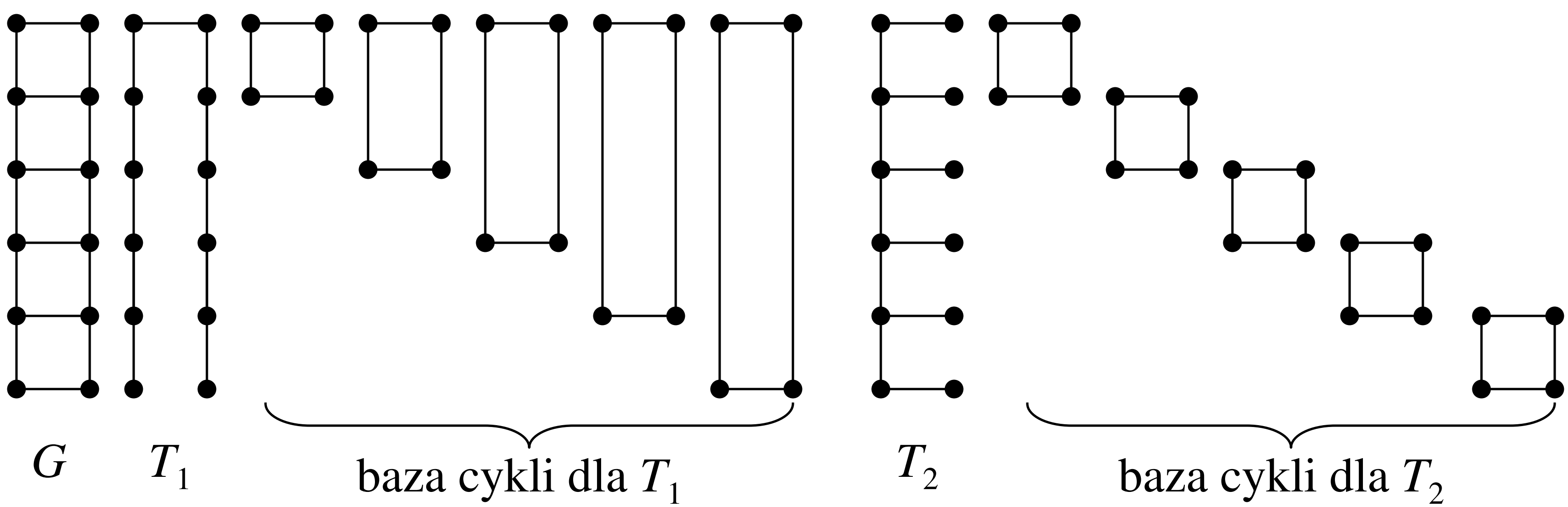
$$\gamma(G_1) + \ldots + \gamma(G_k) = (m_1 + \ldots + m_k) - (n_1 + \ldots + n_k) + k = m - n + k,$$

co kończy dowód, gdyż $\gamma(G) = \gamma(G_1) + \ldots + \gamma(G_k)$.

Baza cykli

Def Niech T będzie lasem spinającym grafu G. *Fundamentalny zbiór cykli* (*baza cykli*) jest to zbiór $\gamma(G)$ różnych cykli grafu G taki, że każdy cykl do niego należący powstał przez dodanie pewnej krawędzi ze zbioru $E(G) \backslash E(T)$.

Przykład Dwa fundamentalne zbiory cykli pewnego grafu, uzyskane za pomocą różnych drzew spinających T_1 i T_2.



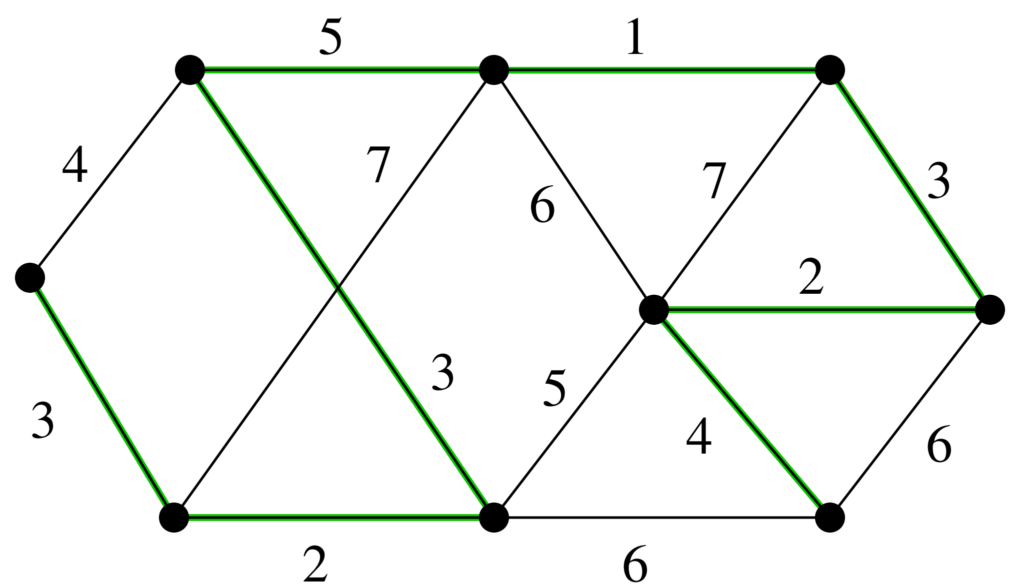
Minimalne drzewa spinające

Niech będzie dany graf G. Zakładamy, że z każdą krawędzią $e_i \in E(G)$ jest skojarzona pewna liczba nieujemna, będąca wagą tej krawędzi. Rozważamy problem szukania drzewa spinającego którego suma wag krawędzi jest minimalna. Przykładem rozwiązania dla tego problem jest algorytm Prima o złożoności $O(n^2)$.

Procedure Prim(G, s)
begin
  $T = (\{ v_i, v_j \}, \{\{v_i, v_j\}\})$, gdzie $\{v_i, v_j\}$ jest krawędzią o
    najmniejszej wadze;
  while $n(T) < n(G)$ **do begin**
    znajdź najkrótszą krawędź $\{u,v\}$ taką, że $u \in V(T)$ oraz $v \notin V(T)$;
    $T := T \cup \{u,v\}$;
  end
end

Minimalne drzewa spinające

Procedure Prim(G, s)
begin
  $T = (\ \{ v_i,v_j\}, \{\{v_i,v_j\}\}\ )$, gdzie $\{v_i,v_j\}$ jest krawędzią o najmniejszej wadze;
  while $n(T) < n(G)$ **do begin**
    znajdź najkrótszą krawędź $\{u,v\}$ taką, że $u \in V(T)$ oraz $v \notin V(T)$;
    $T := T \cup \{u,v\}$;
  end
end



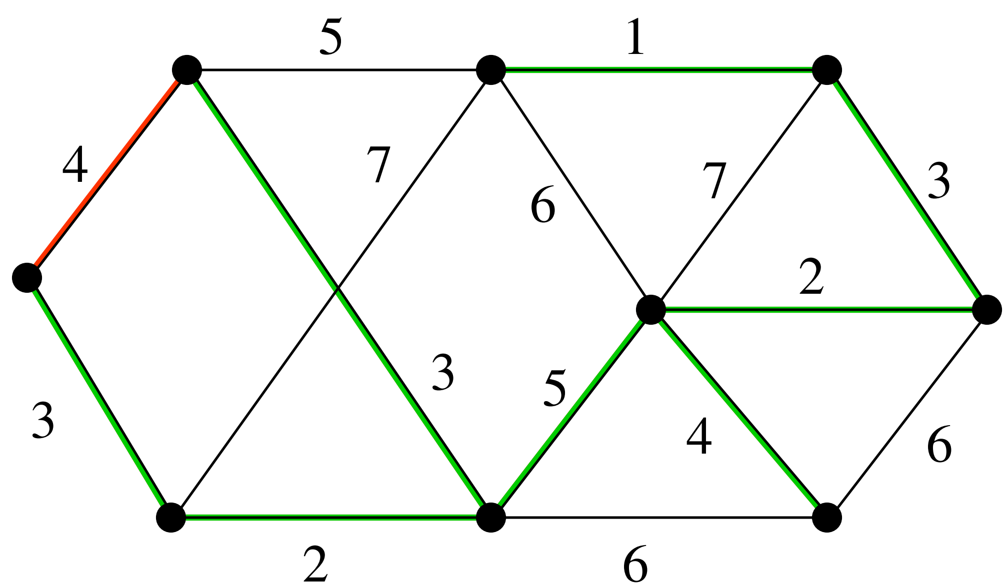
Minimalne drzewa spinające

Algorytm Kruskala jest drugim przykładem efektywnego szukania minimalnego drzewa spinającego. Złożoność tego algorytmu to $O(m \log(m))$.

Procedure Kruskal(G, s)
begin
  posortuj krawędzie $e_1, \ldots, e_m$ niemalejąco wg wag;
  $T = \emptyset$;
  $i := 1$; $j := 0$;
  repeat
    if e_i nie tworzy cyklu w T **then begin**
      $T := T \cup \{e_i\}$;
      $j := j + 1$;
    end;
    $i := i + 1$;
  until $j = n - 1$
end

Minimalne drzewa spinające

Procedure Kruskal(G, s)
begin
posortuj krawędzie e_1, ... , e_m niemalejąco wg wag;
$T = \emptyset$;
$i := 1$; $j := 0$;
repeat
if e_i nie tworzy cyklu w T **then begin**
$T := T \cup \{e_i\}$;
$j := j + 1$;
end;
$i := i + 1$;
until $j = n - 1$
end



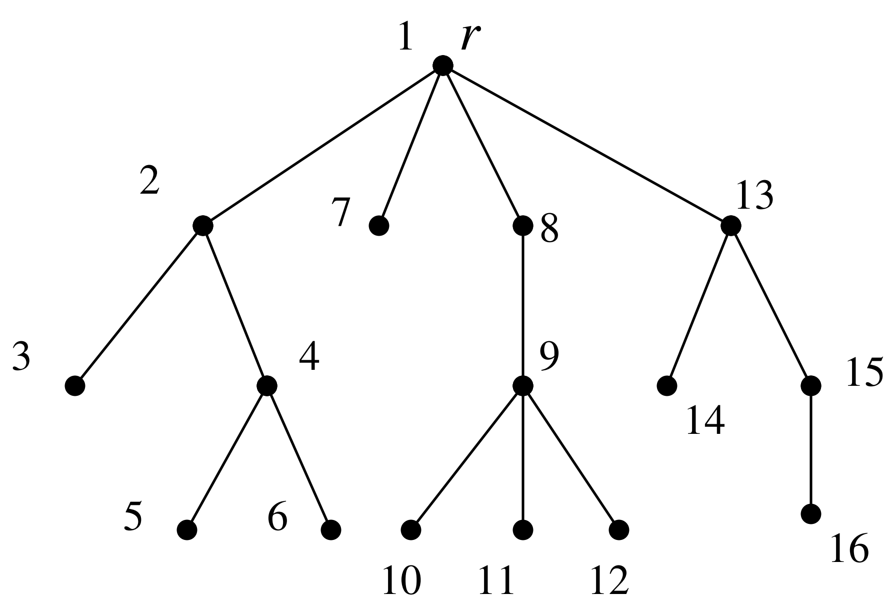
Przybliżone algorytmy dla problemu komiwojażera

- rozważamy problem komiwojażera dla grafów „z nierównością trójkąta”
- DFS – procedura pomocnicza
- algorytmy:
  - algorytm 2-przybliżony
  - algorytm Christofidesa

Algorytm *DFS*

Procedure *DFS*(*G*, *v*)
begin
  odwiedź wierzchołek *v*;
  for każdy nieodwiedzony sąsiad *u* wierzchołka *v* **do**
    DFS(*G*,*u*);
end

Przykład Działanie procedury w przypadku drzewa (wywołanie *DFS*(*T*,*r*))



Algorytm 2-przyblizony

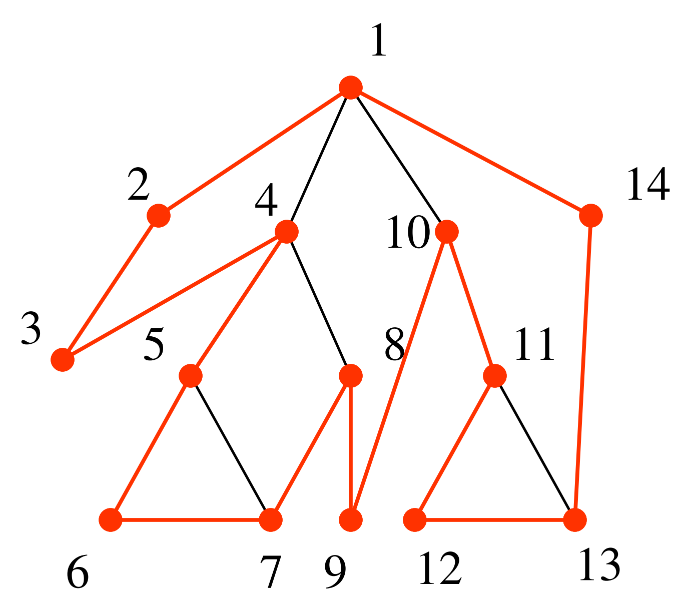
Procedure *A*(*G*)
begin
znajdź minimalne drzewo spinające T grafu G;
DFS(T,v), gdzie v jest dowolnym wierzchołkiem drzewa T;
jeśli $v_1,\ldots,v_n$ są kolejnymi odwiedzonymi wierzchołkami, to utwórz cykl
Hamiltona następująco: $v_1 \to v_2 \to \ldots \to v_{n-1} \to v_{\mathrm{n}} \to v_1$.
return C;
end

Przykład

1) Załóżmy, że znaleziono następujące drzewo spinające pewnego grafu pełnego G

2) kolejność odwiedzania wierzchołków przez *DFS*

3) wyznaczamy cykl Hamiltona



Algorytm 2-przyblizony

Oznaczenia: $W(H) = \Sigma_{e\in E(H)} w(e)$, gdzie H jest podgrafem G.
C_o – najkrótszy cykl Hamiltona w grafie G.
T_o – najlżejsze drzewo spinające grafu G.

Tw. Dla algorytmu A zachodzi $W(A(G))/W(C_o)<2$.
Dowód: Mamy oszacowanie: $W(T_o) < W(C_o)$.
Z nierówności trójkąta wynika, że $W(A(G)) \leq 2W(T_o)$.
Stąd

$$\frac{W(A(G))}{W(C_o)} \leq \frac{2W(T_o)}{W(C_o)} < \frac{2W(C_o)}{W(C_o)} = 2$$

Algorytm Christofidesa

Jest to przykład suboptymalnego algorytmu dla problemu komiwojażera.

- jak poprzednio, G jest grafem pełnym z obciążonymi krawędziami
- czas działania algorytmu to $O(n^3)$
- długość znalezionego cyklu (suma wag jego krawędzi) jest co najwyżej 1.5 razy dłuższa od długości najkrótszego cyklu

Procedure Christofides(G)
begin
  znajdź minimalne drzewo spinające T_o grafu G;
  znajdź zbiór V^{odd} węzłów nieparzystego stopnia w drzewie T_o;
  znajdź w V^{odd} minimalne skojarzenia dokładne M_o^{odd};
  znajdź cykl Eulera w podgrafie indukowanym przez $T_o \cup M_o^{odd}$;
  przekształć cykl Eulera w cykl Hamiltona C_{ch} w grafie pełnym;
end

Oszacowanie dolne

Poniżej podamy pewne oszacowania dotyczące długości optymalnej trasy komiwojażera, które będą potrzebne podczas analizy algorytmu.

Oznaczenia: C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
T_o – minimalne drzewo spinające grafu G
$W(H) = \Sigma_{e \in E(H)} w(e)$, gdzie H jest podgrafem G.

Lemat $W(T_o) < W(C_o)$

Dowód: Niech e będzie dowolną krawędzią cyklu C_o. Podgraf $C_o - e$ jest ścieżką oraz jest również pewnym drzewem spinającym grafu G. Stąd otrzymujemy :

$$W(C_o) > W(C_o - e) \geq W(T_o).$$

Oszacowanie dolne

Oznaczenia: C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
M_o – minimalne skojarzenia dokładne grafu G
$W(H) = \Sigma_{e \in E(H)}\, w(e)$, gdzie H jest podgrafem G.

Lemat $W(M_o) \leq W(C_o)/2$

Dowód: Niech M_1 oraz M_2 będą podzbiorami krawędzi cyklu C_o trawersowanymi w odpowiednio parzystych i nieparzystych krokach. Bez utraty ogólności możemy założyć, że $W(M_1) \leq W(M_2)$. Z równości

$$W(M_1) + W(M_2) = W(C_o)$$
$$2W(M_1) \leq W(M_1) + W(M_2)$$

wynika, że

$$W(M_1) \leq W(C_o)/2.$$

Po uwzględnieniu, że $W(M_o) \leq W(M_1)$ otrzymujemy tezę.

Algorytm Christofidesa

Oznaczenia: T_o – najlżejsze drzewo spinające grafu G
V_o^{odd} – wierzchołki o nieparzystym stopniu w T
M_o^{odd} – minimalne skojarzenia dokładne w V_o^{odd}
C_o^{odd} – najkrótszy cykl Hamiltona w grafie induk. przez V_o^{odd}
C_o – najkrótszy cykl Hamiltona w obciążonym grafie G
C_{ch} – cykl znaleziony przez alg. Christofidesa
C_E – cykl Eulera w grafie ($V(G)$, $E(T_o) \cup E(M_o^{odd})$)

Lemat *Jeśli G spełnia nierówność trójkąta, to* $W(C_{ch}) \leq 3W(C_o)/2$

Dowód: W poprzednich lematach wykazaliśmy, że

$W(T_o) < W(C_o)$ oraz $W(M_o^{odd}) \leq W(C_o^{odd})/2$.

Z nierówności $W(C_o^{odd}) \leq W(C_o)$ wynika, że $W(M_o^{odd}) \leq W(C_o)/2$.

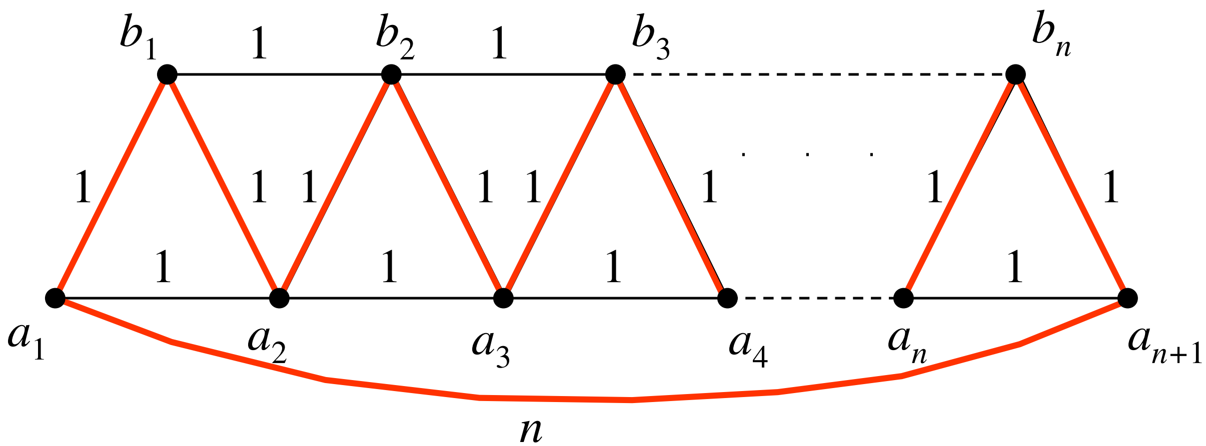
Z faktu, że G spełnia nierówność trójkąta wynika, że $W(C_{ch}) \leq W(C_E)$.

Łącząc powyższe nierówności otrzymujemy:

$$W(C_{ch}) \leq W(C_E) = W(T_o) + W(M_o^{odd}) < W(C_o) + W(C_o)/2 = 3W(C_o)/2.$$

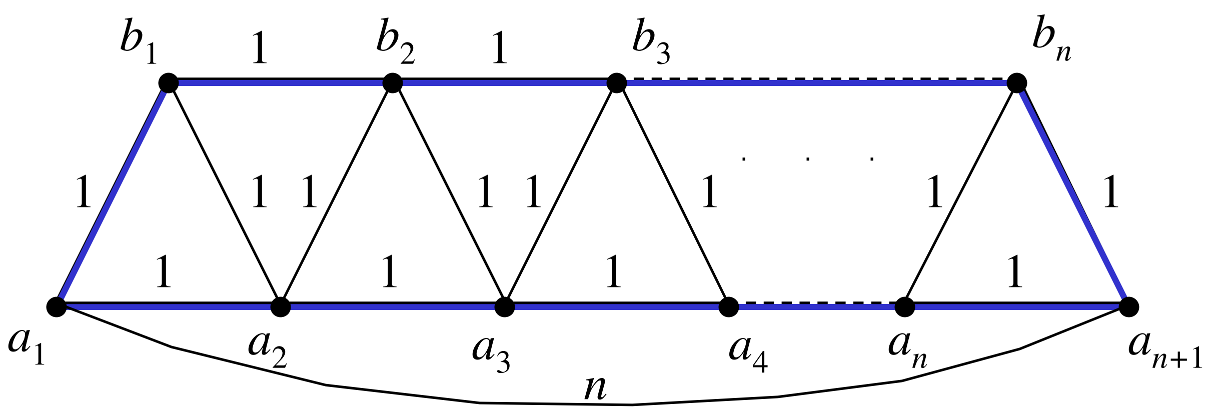
Algorytm Christofidesa

Uwaga Istnieją grafy, dla których współczynnik $W(C_{ch})/W(C_o)$ może przyjmować wartość dowolnie bliską 3/2.



1) minimalne drzewo spinające

2) $W(C_{ch}) = 3n$



3) rozwiązanie optymalne $3W(C_o) = 2n + 1$