Maksymalny przepływ

Założenia:

- sieć przepływową (np. przepływ cieczy, prądu, danych w sieci itp.) będziemy modelować za pomocą grafów skierowanych
- łuki grafu odpowiadają kanałom
- wierzchołki to miejsca połączeń kanałów
- waga łuku odpowiada przepustowości kanału; $c(e)>0$ dla każdego łuku e,
- wyróżniamy dwa punkty: s – źródło oraz t – ujście,
- w niniejszym wykładzie digraf $\equiv$ sieć

Wstęp

Def. *Przepływem* nazywamy funkcję f:$V\times V\to$R spełniającą warunki:

1. dla wszystkich $u,v \in V$ $f(u,v) \le c(u,v)$ (przepustowość)
2. dla wszystkich $u,v \in V$ $f(u,v) = -f(v,u)$ (skośna symetryczność)
3. dla każdego $u \in V \setminus \{s,t\}$
$$\sum_{v\in V} f(u,v) = 0$$ (zachowanie przepływu)

Wartość przepływu to $|f| = \sum_{v\in V} f(s,v)$.

Uwaga Przyjmujemy, że $f(X,Y) = \sum_{x\in X} \sum_{y\in Y} f(x,y)$, gdzie $X \subseteq V$, $Y \subseteq V$.

Def. Jeśli f_1, f_2 są przepływami dla tej samej sieci, to sumę przepływów f_1, $+ f_2$ definiujemy jako

$$(f_1 + f_2)(u,v) = f_1(u,v) + f_2(u,v).$$

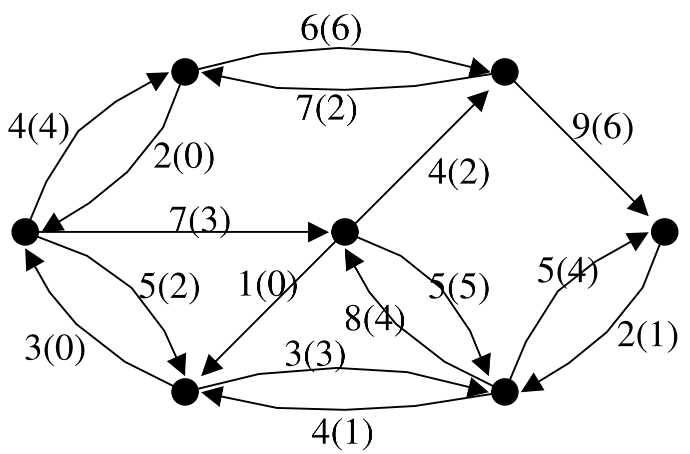
Sieć residualna

Def. Niech będzie dana sieć G z funkcją wagową c oraz przepływ f dla G. *Przepustowość residualną* odpowiadającą f definiujemy jako

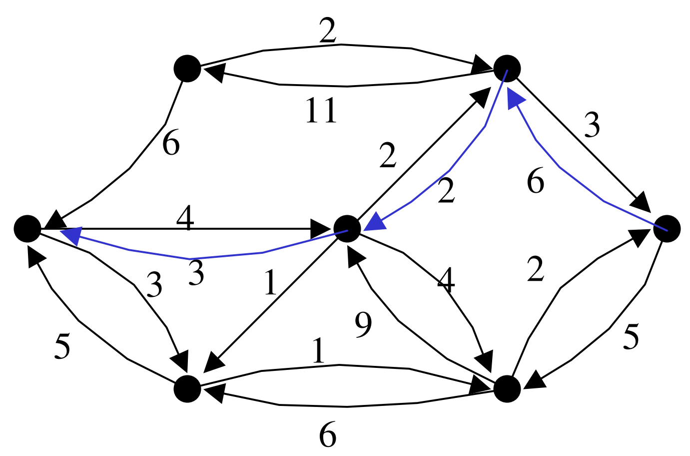
$$c_f(u,v) = c(u,v) - f(u,v)$$

dla wszystkich wierzchołków u,v grafu G. *Sieć residualna* dla f to digraf G_f taki, że $V(G_f) = V(G)$ oraz $E(G_f) = \{\ u,v \in V\times V\colon c_f(u,v)>0\ \}$.

Przykład



Graf G (etykieta krawędzi to $c(f)$)



Graf G_f (etykiety to c_f)

Suma przepływów

Lemat *Niech G, G_f będą odpowiednio siecią oraz siecią residualną utworzoną na podstawie przepływu f oraz niech f' będzie przepływem w sieci G_f. Wówczas f + f' jest przepływem w G o wartości* $|f| + |f'|$.

Dowód: Sprawdzamy czy $f + f'$ jest przepływem:

1. (przepustowość):

$$(f + f')(u,v) = f(u,v) + f'(u,v) \leq f(u,v) + c_f(u,v) =$$
$$= c(u,v) - c_f(u,v) + c_f(u,v) = c(u,v)$$

2. (skośna symetryczność):

$$(f + f')(u,v) = f(u,v) + f'(u,v) = -f(v,u) - f'(v,u) =$$
$$= -(f(v,u) + f'(v,u)) = -(f + f')(v,u)$$

3. (zachowanie przepływu):

$$\sum_{v \in V}(f + f')(u,v) = \sum_{v \in V}(f(u,v) + f'(u,v)) = \sum_{v \in V} f(u,v) + \sum_{v \in V} f'(u,v) = 0 + 0 = 0$$

Wartość:

$$|f + f'| = \sum_{v \in V}(f + f')(s,v) = \sum_{v \in V} f(s,v) + \sum_{v \in V} f'(s,v) = |f| + |f'|$$

Ścieżka powiększająca

Def. *Ścieżką powiększającą* nazywamy dowolną ścieżkę od źródła s do ujścia t w sieci residualnej G_f. *Przepustowością residualną* ścieżki P jest liczba

$$c_f(P) = \min\{\ c_f(u,v) : (u,v) \in E(P)\ \}$$

Lemat *Niech P będzie ścieżką powiększającą dla przepływu f i sieci residualnej G_f. Wówczas funkcja*

$$f_P(u,v) = \begin{cases} c_f(P) & gdy\ (u,v) \in E(P) \\ -c_f(P) & gdy\ (v,u) \in E(P) \\ 0 & else \end{cases}$$

jest przepływem w sieci G_f takim, że $|f_P| = c_f(P) > 0$. Ponadto dla przepływu $f' = f + f_P$ zachodzi $|f'| > |f|$.

Wniosek *Przy założeniach z powyższego lematu mamy, że $f + f_p$ jest przepływem o wartości większej niż $|f|$.*

Przekroje

Def.

- *Przekrojem* w digrafie G nazywamy parę S, T taką, że $s \in S$, $t \in T$ oraz $S \cup T = V$ i $S \cap T = \varnothing$
- jeśli f jest przepływem, to *przepływ netto* przez przekrój (S,T) to $f(S,T)$,
- *przepustowość* przekroju (S,T), to $c(S,T)$.

Lemat *Jeśli f jest przepływem w digrafie G, ze źródłem s i ujściem t, (S,T) jest przekrojem, to przepływ netto przez przekrój (S,T) wynosi* $f(S,T) = |f|$.

Wniosek *Wartość dowolnego przepływu f w sieci G nie może być większa niż przepustowość dowolnego przekroju w G.*

Przekroje

Tw. *Niech f będzie przepływem w digrafie G ze źródłem s i ujściem t. Poniższe warunki są równoważne:*

1. *przepływ f jest maksymalny,*
2. *sieć residualna G_f nie zawiera ścieżek powiększających,*
3. *dla pewnego przekroju (S,T), $|f| = c(S,T)$.*

Szkic dowodu:

(1⇒2) Jeśli założymy, że ścieżka powiększająca istnieje, to z wcześniejszego lematu otrzymujemy, że można otrzymać większy przepływ.

(2⇒3) Definiujemy zbiór S będący zbiorem wierzchołków osiągalnych z s w sieci residualnej G_f. Wówczas $(S,V(G)\setminus S)$ jest przekrojem.

(3⇒1) Wynika z poprzedniego wniosku.

Algorytm

G – digraf
s – źródło
t – ujście

procedure Fulkerson-Ford(G, s, t)
begin
  for each $(u,v) \in E(G)$ **do begin**
    $f(u,v) := 0;$
    $f(v,u) := 0;$
  end;
  while istnieje ścieżka powiększając P w digrafie G_f **do begin**
    $c_f(P) := \min\{\ c_f(u,v) \colon (u,v) \in E(P)\ \};$
    for each $(u,v) \in E(P)$ **do begin**
      $f(u,v) := f(u,v) + c_f(P);$
      $f(v,u) := f(v,u) - c_f(P);$
    end;
  end
end

Złożoność

- czas wyszukiwania ścieżki w sieci residualnej to $O(|E(G_f)|) = O(|E(G)|)$;
- każda iteracja pętli while jest więc wykonywana w czasie $O(|E(G)|)$;
- jeśli przepustowości są liczbami całkowitymi, to pętla while wykonuje co najwyżej $|f_{max}|$ razy, gdzie f_{max} jest wartością maksymalnego przepływu;
- jeśli przepustowości są całkowite, to złożoność wynosi $O(|E(G)|\,|f_{max}|)$;
- jeśli nie ma żadnych ograniczeń na przepustowości (mogą być dowolnymi liczbami rzeczywistymi), to możliwe jest, że algorytm będzie działał w nieskończoność;

Implementacja Edmondsa-Karpa

Uwaga Aby poprawić złożoność algorytmu, w każdej iteracji pętli while, będziemy szukać najkrótszej ścieżki powiększającej (tutaj i w dalszej części tego wykładu długość ścieżki to liczba należących do niej łuków). Niech $d_f(u,v)$ oznacza odległość pomiędzy wierzchołkami u,v (=długość najkrótszej ścieżki pomiędzy u i v) w sieci residualnej G_f.

Lemat *Dana jest sieć G ze źródłem s i ujściem t. Jeśli w każdym przebiegu pętli while szukana jest najkrótsza ścieżka powiększająca w sieci G_f, to dla każdego wierzchołka $v \in V(G)\setminus\{s,t\}$ odległość v od s w sieci residualnej G_f nie maleje.*

Złożoność

Tw. *Dla danej sieci G ze źródłem s i ujściem t, liczba powiększeń przepływu wynosi* $O(|V(G)||E(G)|)$.

Szkic dowodu:

- łuk (u,v) na ścieżce powiększającej P jest *krytyczny* jeśli przepustowość residualna ścieżki P jest równa przepustowości residualnej $\{u,v\}$;
- po zwiększeniu przepływu łuk krytyczny usuwany jest z sieci residualnej;
- z faktu, że ścieżka P jest najkrótsza wynika, że $d_f(s,v) = d_f(s,u) + 1$;
- łuk (u,v) pojawia się znowu w sieci residualnej, gdy pojawi się inna ścieżka powiększająca, która zawiera (v,u)
- jeśli aktualnym wówczas przepływem jest f', to $d_{f'}(s,u) = d_{f'}(s,v) + 1$;
- z dwóch powyższych równości i wcześniejszego lematu wynika, że $d_{f'}(s,u) \geq d_f(s,u) + 2$;
- odległość u od s rośnie o co najmniej 2 od chwili, gdy (u,v) był krytyczny do chwili, gdy (u,v) staje się krytyczny ponownie
- stąd, każdy łuk może być krytyczny co najwyżej $O(|V(G)|)$ razy

Najtańszy przepływ

Założenia:

- dany jest graf skierowany G;
- mamy wyróżnione wierzchołki s i t – źródło i ujście;
- z każdym łukiem (u,v) w G są skojarzone dwie etykiety: $c(u,v)$ przepustowość, $d(u,v)$ – koszt;

Zadanie:

- znaleźć najtańszy przepływ w sieci G, czyli znaleźć przepływ f o zadanej wartości F, który minimalizuje sumę

$$\sum_{(u,v)\in E(G)} d(u,v) f(u,v)$$

Szkic algorytmu

- w digafie G szukamy najtańszą drogę z s do t;
- przez najtańszą drogę rozumiemy najkrótszą względem funkcji wagowej d;
- przesyłamy tą drogą możliwie najwięcej jednostek przepływu;
- jeśli liczba przesłanych jednostek przekracza F, to najtańszy przepływ został znaleziony i obliczenia są przerywane;
- jeśli liczba przesłanych jednostek jest mniejsza niż F, to modyfikujemy odpowiednio sieć;
- powtarzamy powyższe kroki dla zmodyfikowanej sieci;

Modyfikacja sieci

- dana jest sieć G, funkcje c,d oraz pewien przepływ f,
- zmodyfikowana sieć ma identyczny zbiór wierzchołków i krawędzi, jej nowe funkcje oznaczamy odpowiednio przez c',d',

Niech łuk (u,v) ma niezerowy przepływ w sieci G. Wówczas:

1. $c'(v,u) := f(u,v)$ [dodanie „pozornego” łuku]
2. $d'(v,u) := -d(u,v)$
3. $c'(u,v) := c(u,v) - f(u,v)$ jeśli $f(v,u)=0$ i $c(u,v) > f(u,v)$
4. $d'(u,v) := d(u,v)$ jeśli $f(v,u)=0$ i $c(u,v) > f(u,v)$
5. $c'(u,v) := 0$ jeśli $f(v,u)=0$ i $c(u,v) = f(u,v)$
6. $d'(u,v) := +\infty$ jeśli $f(v,u)=0$ i $c(u,v) = f(u,v)$

Uwaga Dalej pisząc G' mamy na myśli digraf G z funkcjami wagowymi c',d'.

Algorytm Busackera i Gowena

procedure BG(*G, c, d*)
begin
  G' := *G*;
  while ($|f| < F$) **and** (istnieje droga z *s* do *t* w *G'*) **do begin**
    znajdź najkrótszą ścieżkę *P* łączącą *s* i *t* w *G'*;
    if ścieżka *P* nie istnieje **then**
      „koniec";
    zmodyfikuj przepływ *f* wykorzystując *P* przesyłając maksymalną liczbę jednostek wzdłuż *P* (jeśli uzyskany przepływ jest większy niż *F*, to przesyłamy jedynie taką liczbę jednostek, która daje przepływ o wartości *F*;
    wyznacz zmodyfikowaną sieć *G*';
  end
end

Złożoność: $O(n^2F)$