データ構造とアルゴリズムII

第11回
最大フロー

536

Johnsonのアルゴリズム

- 疎なグラフ，例えば$|E| \leqq O(|V|\lg|V|)$が仮定できる場合に向いている
- 隣接リスト表現を仮定する.
- 実行時間は$O(V^2 \lg V + VE)$.
- 上記の仮定の下で，Floyd-Warshallアルゴリズムよりも漸近的に高速

537

Johnsonのアルゴリズムのアイデア

- 辺重みが全部非負なら，Dikstraのアルゴリズムをすべての点について実行すれば最短路が得られる.
- 負辺があるが，負閉路を持たない場合は，再重み付けを行う.
- すなわち，各頂点vに関して h(v) を定義し，w'(u, v)=w(u, v) + h(u) - h(v)とする.
これにより最短経路は変化しないことに注意.
- 新しい重みの元での最短経路長 δ'(u, v)から，元の問題の最短経路長は以下で与えられる:
δ(u, v)=δ'(u, v) - h(u) + h(v).

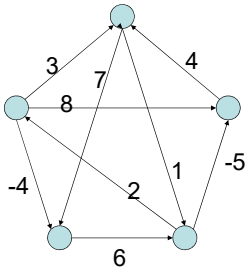
538

Johnsonのアルゴリズム

```
Johnson(G)
{ compute G', where V[G']=V[G]∪{s} and
   E[G']=E[G]∪{(s,v):v∈V[G]} s.t. w(s,v)=0.
  if Bellman-Ford(G',w,s)=False
     then print "∃ a neg-weight cycle"
  else for each vertex v ∈V[G']
          set h(v)=δ(s,v)
              computed by Bellman-Ford algo.
       for each edge (u,v)∈E[G']
          w'(u,v)=w(u,v)+h(u)-h(v)
       for each vertex u ∈V[G]
          run Dijkstra(G,w',u) to compute δ'(u,v)
       for each v ∈V[G]
          d_uv=δ'(u,v)-h(u)+h(v)
   return D
}
```

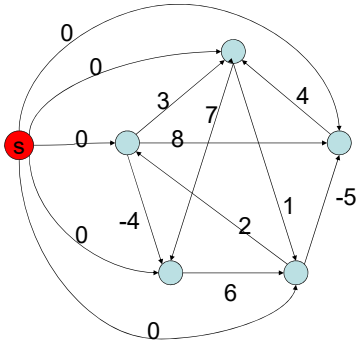
539

Johnsonのアルゴリズムの実行例



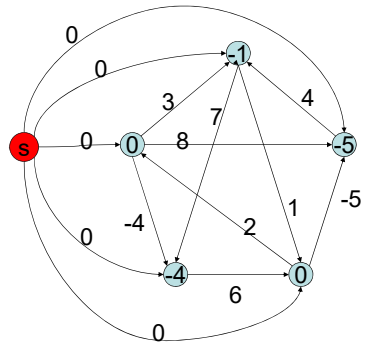
540

Johnsonのアルゴリズムの実行例



541

Johnsonのアルゴリズムの実行例

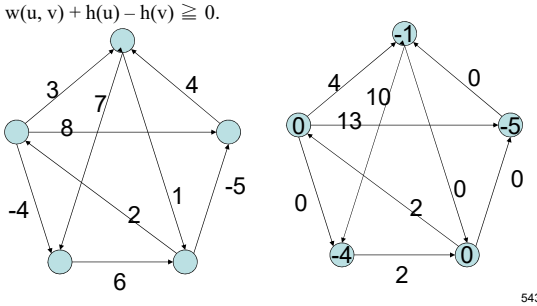


542

Johnsonのアルゴリズムの実行例

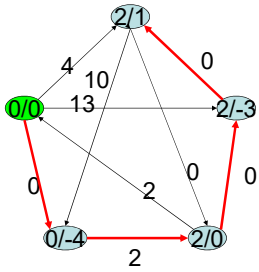
三角不等式より，u, vに関して，
h(u) + w(u, v) ≧ h(v), よって，
w(u, v) + h(u) – h(v) ≧ 0.

再重み付け



543

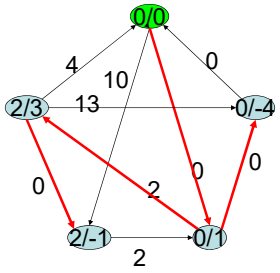
Johnsonのアルゴリズムの実行例



赤が最短経路.

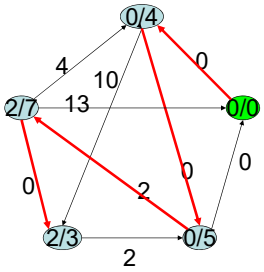
544

Johnsonのアルゴリズムの実行例



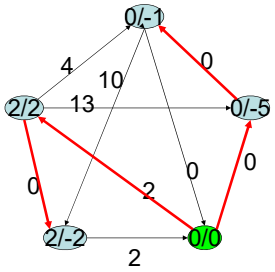
545

Johnsonのアルゴリズムの実行例



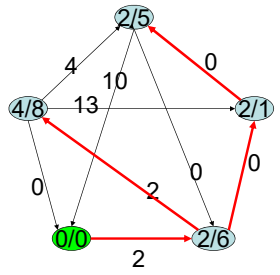
546

Johnsonのアルゴリズムの実行例



547

Johnsonのアルゴリズムの実行例



548

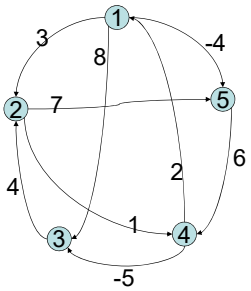
Johnsonのアルゴリズムの計算量

- Bellman-Fordの実行はO(|V||E|).
- Dijkstraを|V|回実行.
  - 凝ったヒープ (Fibonacci heap)を使えば 総計O(|V|^{2}log|V|+|V||E|) .
  - 普通のBinary heapだと，総計O(|V||E|log|V|).
- |E|が少なければ（≦|V|lg|V|），Warshall-Floydより良い.

549

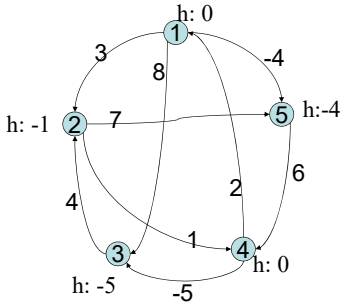
演習：Johnsonのアルゴリズム

- 右のグラフに関してJohnsonのアルゴリズムを適用せよ（最短経路は，1からの経路のみを求めるのみで良い）



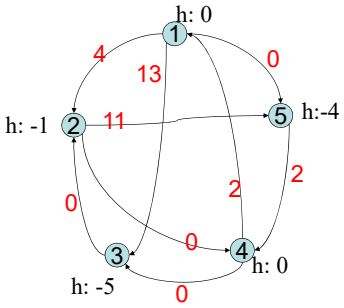
550

演習：Johnsonのアルゴリズム



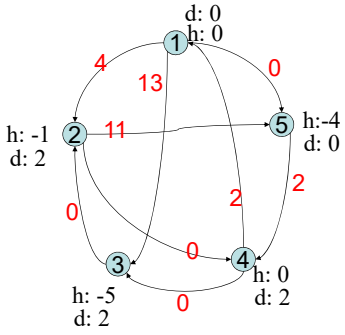
551

演習：Johnsonのアルゴリズム



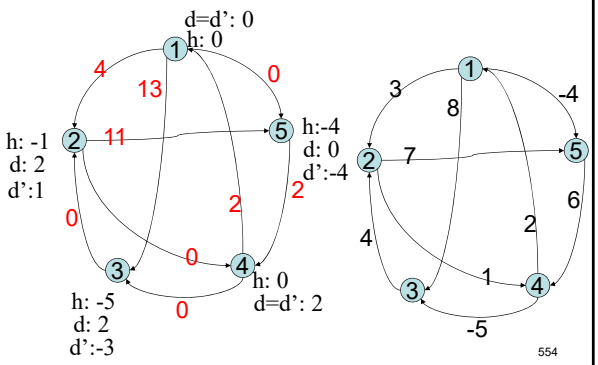
552

演習：Johnsonのアルゴリズム



553

演習：Johnsonのアルゴリズム



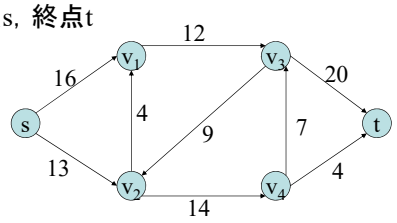
554

26. 最大フロー

555

フローネットワーク

- フローネットワーク：G=(V,E)は有向グラフで，各辺(u, v)∈E には容量 c(u, v)>0が定義される (u, v間に辺がなければc(u, v)=0と仮定)
- 始点s，終点t



556

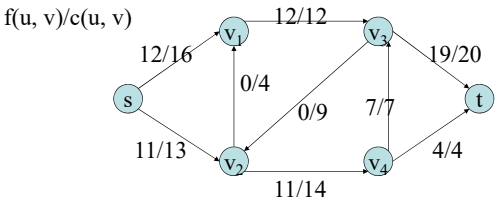
フロー

- フロー／流量は関数f: V × V→Rで定義され，任意の u, v に関して以下の条件を満たす：
  - 容量制限: f(u, v)≤c(u, v)
  - フロー保存則: すべてのu∈V-{s,t}に関して，$\Sigma_{v\in V}f(v, u)= \Sigma_{v\in V}f(u, v)$が成立.
- $|f|=\Sigma_{v\in V}f(s, v)-\Sigma_{v\in V}f(v, s)$を フロー f の値と呼ぶ（絶対値とは無関係）.

557

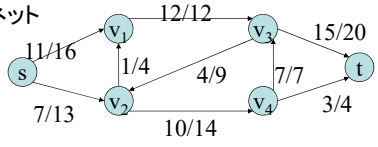
最大流量問題

- フローの値が最大となる f を求める.

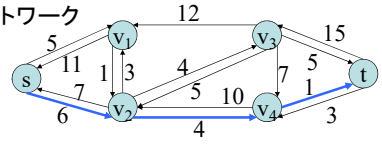


558

フローネットワーク



残余ネットワーク



増加可能パス

559

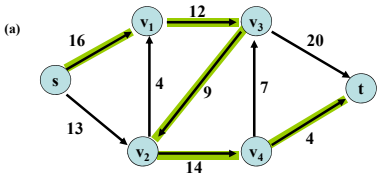
Ford-Fulkerson法

- 増加可能パスが存在する限り，残余ネットワークにフローを加え続ける．
- 増加可能パスがなければ終了．

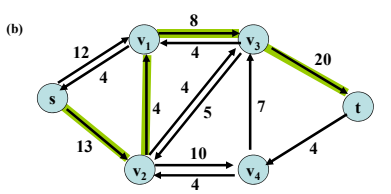
560

```
Ford-Fulkerson(G,s,t)
{ for each edge (u,v)∈E[G]
    do f[u,v]←0
       f[v,u]←0
  while ∃ path p from s to t on Gf
    do cf(p)←min{cf(u,v):(u,v) is in p}
       for each (u,v) in p
          if (u, v) ∈ E[G]
             f[u,v]←f[u,v]+ cf(p)
          else f[v,u]←f[u,v]-cf(p)
  return f
}
```

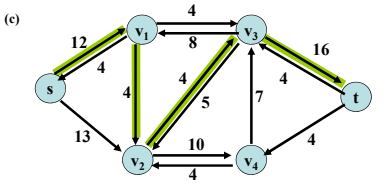
561



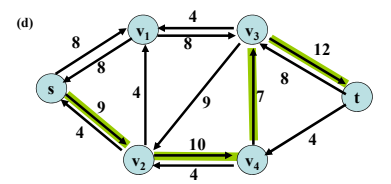
562



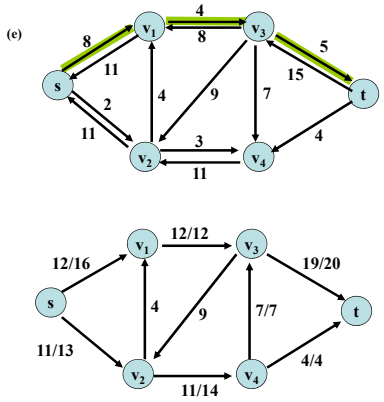
563



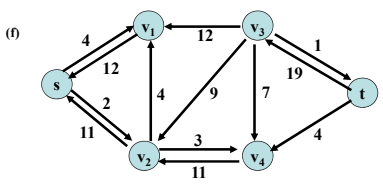
564



565



566



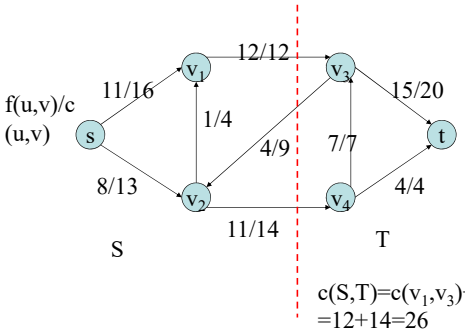
567

最大フローと最小カット

- フローネットワークG=(V, E)のカット (cut) (S, T)は，VのSとTの分割でs∈S及びt∈Tを満たすもの.
- カット (S, T) とフロー f に関して，カットと交差する純フロー f(S, T) は以下で与えられる: $\sum_{u\in S}\sum_{v\in T} f(u,v) - \sum_{u\in S}\sum_{v\in T} f(v,u)$.
- Cut (S,T)の容量 (Capacity)，c(S, T)は以下で与えられる: $\sum_{u\in S}\sum_{v\in T} c(u,v)$.
- ネットワークの最小カットは，このネットワークのすべてのカット中で容量が最小のもの.

568

Cutの例



c(S,T)=c(v_1,v_3)+c(v_2,v_4)
=12+14=26

569

補題26.4

- 任意のカット(S, T)とフローfに関して，(S, T)と交差する純フローは，フローの流量|f|に等しい.

570

補題26.4の証明

フロー保存則より，s,t 以外の任意の頂点 u に関して，
$\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u) = 0$ が成立．よって，
$\sum_{u\in S-\{s\}}(\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u)) = 0$ が成立．

$$
\begin{aligned}
|f| &= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) \\
&= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{u\in S-\{s\}} [\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u)] \\
&= \sum_{u\in S}[\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u)] \\
&= \sum_{u\in S}[\sum_{v\in T} f(u,v) - \sum_{v\in T} f(v,u)] + \sum_{u\in S}[\sum_{v\in S} f(u,v) - \sum_{v\in S} f(v,u)] \\
&= \sum_{u\in S}[\sum_{v\in T} f(u,v) - \sum_{v\in T} f(v,u)] \\
&= f(S,T)
\end{aligned}
$$

571

系26.6

- フローネットワークの流量は任意のカットの容量で上から抑えられる.
- 証明：容量制限から明らか

572

最大フロー＝最小カット

- 定理26.6: 最大フロー最小カット定理
- 以下の条件は等価

1. fはGにおける最大フローである.
2. 残余ネットワークG_fは増加可能経路を含まない.
3. Gのあるカット (S, T) に関して|f| = c(S, T) である.

573

証明

- 1⇒2: 1だか2でない，すなわち，最大フローであるのに，増加可能パスがあるとすると，明らかに矛盾.
- 2⇒3: 残余ネットワークでは，sからtへの経路が存在しない. sから到達可能な頂点集合をS, 残りをTとすると，これはカットになっている. u∈S, v∈Tに関して，(u, v) ∈Eなら，この辺は容量一杯まで流れているはず（そうでないとvがsから到達可能）. また，(v, u)∈Eなら，この辺のフローは0でないと矛盾. 補題26.4より，(S, T)と交差する純フローは，フローの流量に等しい.
- 3⇒1: 系26.5より，すべてのフローの流量はカットの容量で上から抑えられる. よって，カットの容量に等しいことは，フローが最大であることの十分条件である.

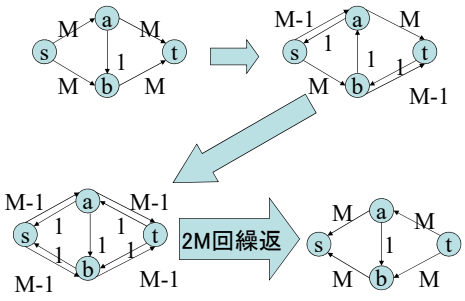
574

Ford-Fulkerson法の計算量

- 増加可能パスを求める方法に依存する.
- 良くない方法を使うと停止しないこともある（辺容量が無理数の場合）.
- 容量が整数なら有限の繰り返しで停止する.
- 幅優先探索を用いて，最短経路の増加可能パスを求めた方が良い（Edmonds-Karpアルゴリズムと呼ばれる）.
- この場合のフロー増加操作の総数は高々O(VE) となる.

575

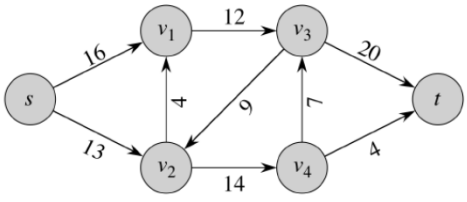
最短でない増加可能パス



576

演習: Ford-Fulkerson

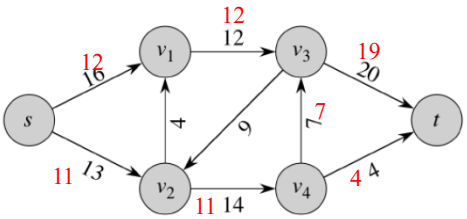
- 以下のグラフの最大フローを求めよ



577

演習: Ford-Fulkerson

- 以下のグラフの最大フローを求めよ



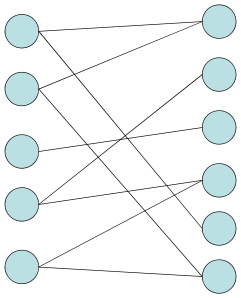
578

二分グラフの最大マッチング (26.3)

- 二分グラフは，頂点集合VがLとRの二つの集合に分割され，任意の辺 (u, v) は，LとRの頂点を結ぶ（L間，R間の辺は存在しない）.

579

二分グラフの例



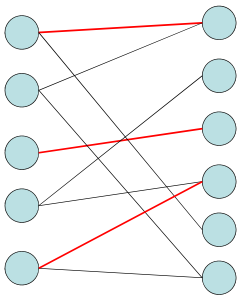
580

マッチング

- すべての頂点vに関して，vに接続する辺が高々一つしかない辺集合をマッチングと呼ぶ.
- 最大マッチングは，要素数が最大のマッチングである.

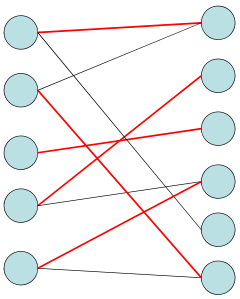
581

マッチングの例



582

最大マッチングの例



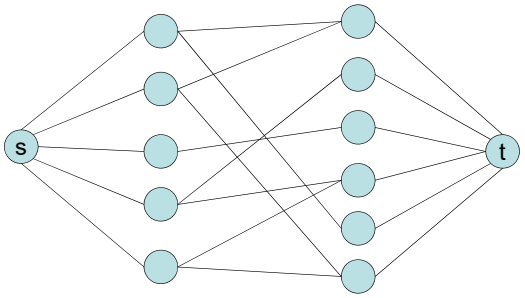
583

最大フローとしての表現

- 新しい頂点sとtを導入
- sから左側の頂点に容量1の辺を張る
- 右側の頂点からtに容量1の辺を張る

584

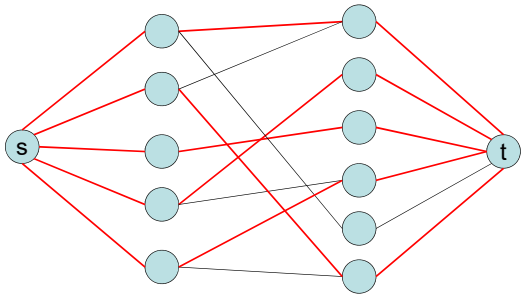
最大フローとしての表現



L R

585

最大フロー＝最大マッチング



L R

586

定期試験に関して

- 来週の7月25日（火）
- 場所は，この部屋ではなく，工学部第4講義室
- 内容は4〜5題ぐらい
- 動的計画法，貪欲法，ならし解析，幅／深さ優先探索，最短路，最大フロー等
- 昨年度までとは（多分）傾向が変わるので注意

587