プログラム設計とアルゴリズム
第10回 (11/29)

早稲田大学高等研究所　講師
福永津嵩

(前回の復習)最短路問題

- 重みなしグラフの場合、幅優先探索により最短路を得る事が出来た。
  一方、重みつきグラフの場合は別のアルゴリズムが必要となる。

- 有向グラフと頂点sが与えられた時、sから各頂点への最短路
  (重みの和が最小の路)を求める問題(単一始点最短路問題)を考える。

図14.1

(前回の復習) DAGの最短路問題

- ある種のDAG(有効非巡回グラフ)については、既に最短路問題を解いている。

図5.2

- これは数字が小さい方から順に最短路長を求める動的計画法で答えを得ることが出来るのであった。

- この事が成り立つのは、辺e = (u, v)に対して緩和処理を行う時、頂点uについてはd[u]が最短路になっている事が原因である。

(前回の復習) DAGの最短路問題

- 逆に言えば、DAGにおいて、頂点uについてはd[u]が最短路になっているならば、同様に動的計画法を行うことで最短路問題を解ける。

- これは、グラフに対してトポロジカルソートを行い、得られた頂点順に動的計画法を行う事と等しい。

- トポロジカルソートの計算量は$O(|V|+|E|)$であった。頂点順の緩和処理も$O(|V|+|E|)$である。

図13.13

- よって計算量は$O(|V|+|E|)$

(前回の復習) ベルマン・フォード法

- 負の辺及び閉路も含む一般的な有向グラフに対して最短路を求める。
(ただし、まず負閉路がない場合を考える)

- ベルマンフォード法とは次のようなアルゴリズムである。

1. 全ての辺について緩和処理を行う。

2. 緩和処理によってd[v]の更新が行われなくなるまで1.を繰り返す。

- 1.の計算量はO(|E|)であり、2.で行われる繰り返し回数は高々|V|-1回なので、この計算量はO(|V||E|)となる。

(前回の復習) ベルマン・フォード法

図14.7

(前回の復習) ダイクストラ法

1. 既に最短路が求められている頂点の集合をSとする。
   初期値として始点sをSへ移動し、sから出る辺について緩和処理を行う。

2. まだSに含まれていない頂点のうち、d[v]が最小になるvを探索し、
   これをSへと移動する。

3. vから出る辺について緩和処理を行う。
   (全ての辺について緩和処理を行う必要がない。)
   全ての頂点がSに含まれた場合にはアルゴリズムを終了し、
   そうでない場合は2へ戻る。

(前回の復習)ダイクストラ法

図14.9

(前回の復習) 全点対間最短路問題

- 単一始点の最短路長ではなく、全頂点対での最短路長を考える。

- ベルマンフォード法を全頂点を始点として適応してもよいが、
  その場合の計算量はO(|V|²|E|)となる。ここでは、O(|V|³)となる
  フロイド・ワーシャル法を紹介する。

フロイド・ワーシャル法における動的計画法

(前回の復習) まとめ

表14.1

- 全点対間最短路問題は、フロイド・ワーシャル法によって$O(|V|^3)$で計算可能である。

第十五章
グラフ(3):
最小全域木問題

最小全域木問題

- 重みつき無向グラフG=(V, E)が与えられているとする。
  Gの部分グラフであって木であり、かつGの全ての頂点を含むものを
  全域木と呼ぶ。
- 全域木Tの重みは、それに含まれる辺の重みの総和とする。
  この時、最小全域木問題とは、重みが最小の全域木を求める問題である。

図15.1

クラスカル法

- 単純な貪欲法によって、最小全域木を求めることが可能である。

最小全域木を求めるクラスカル法

クラスカル法

図15.2

クラスカル法の計算量

- 辺のソートにかかる計算量はO(|E|log|E|)である。
  なお、|E|は高々|V|²であるため、log|E|は高々2log|V|である。
  よって、O(|E|log|V|)とも書ける。

- 辺を追加したときにサイクルを形成するかの判定については、
  Union-Findを利用することで効率的に実現出来る。

(復習) Union-Find

- グループ分けを管理するデータ構造であり、次の処理を行う
  事が出来る。

  ・issame(x, y): x, yが同じグループに属するかどうかを調べる
  ・unite(x, y): xが属するグループと、yが属するグループを併合する。

- 右の例に対しては、

  issame(0, 4) = true
  issame(3, 5) = true
  issame(2, 6) = false

図11.1左

クラスカル法の計算量

- Tに新しい辺e = (u, v)を追加する際、頂点u, vをUnion-Findのデータ構造上でuniteしておく。

- その後、別の辺e' = (u', v')を考えた時に、u', v'が同一のグループに所属しているならば、その辺e'を追加することでサイクルが形成される。よって破棄する。

- 1回の判定にかかる計算量は$\alpha(|V|)$であるから、全ての枝を判定するとその計算量はO(|E|$\alpha(|V|)$)

- よって全体的な計算量はO(|E|log|V|)ととなる。

グラフの基本用語:カット

- グラフG=(V, E)のカットとは、頂点集合XとYへのVの分割を意味する。ただし、X∪Y = V、X∩Y = ϕ とする。

- XとYにまたがる辺をカット辺と呼び、カット辺全体の集合をカットセットと呼ぶ。

図15.3

基本サイクル

・ ある全域木Tにおいて、Tに含まれない辺eを加えると、eとTで
一つのサイクルが形成される。これをTとeに関する基本サイクルと呼ぶ。

図15.4

基本サイクル

- 基本サイクルに含まれる辺を一つ取り除くと、新たな全域木T'ができる

図15.5

基本サイクル

- 最小全域木Tが与えられた時、Tに含まれない辺eについて
  基本サイクルを考えると、そのサイクル内に含まれる辺のうち、
  辺eは最も重みが大きいことがわかる。

- なぜなら、eを含めてサイクルに含まれる別の辺fを取り除いた
  全域木T'を考えると、Tは最小全域木であるから、

$$w(T) \leq w(T') = w(T) + w(e) - w(f)$$

よって、w(e) >= w(f) となる。

基本カットセット

- 全域木TとTに含まれる辺eについて、eを取り除いた時によって
  分割される部分木の頂点集合をXとYとする。

- この時、XとYはカットとなる。そのため、XとYのカットセットを、
  TとEに関する基本カットセットと呼ぶ。

図15.6

基本カットセット

- 基本カットセットも基本サイクルと似たような性質が成り立つ。

- すなわち、基本カットセットに含まれる辺fを取り出して、eを取り除いてからfを加えると、それは全域木となる。

- また、最小全域木においては、T, eに関する基本カットセットをCとすると、Cに含まれる辺のうちeは重みが最小の辺である。

クラスカル法の正当性の証明

- G = (V, E)が与えられた時、Gの全域木Tにおいて、

  A: Tは最小全域木である。
  B: Tに含まれない任意の辺eに対して、Tとeにおいてeのにおいてeの重みは最大である
  が実は成立する。

- A→Bは既に示しており、またクラスカル法で作られる全域木がBである事は明らかである。

- よって、B→Aを示せば、クラルカル法が正当であることが証明される。

全域木間での辺の交換

全域木間での辺の交換

図15.8

クラスカル法の正当性の証明

- 全域木間で辺を交換していくことで、SをTに近づけていくことが出来る。つまり、「Tに含まれるがSには含まれない辺の本数」をSとTの距離とすると、交換によって距離が１ずつ減っていき、0になった時SはTになっている。

- クラスカル法で作成した全域木をT、その他の任意の全域木をSとし、SをTに近づけていく。この時、

$$w(S) \geq w(S') \geq w(S'') \geq \cdots \geq w(T)$$

よって、任意のSに対してw(S) >= w(T)であり、Tは最小全域木となる。

第十六章
グラフ(4):
ネットワークフロー

辺連結度

- ２頂点sからtへのパスにおいて、辺を共有しないs-tパスの最大値を辺連結度と呼ぶ。また、互いに辺を共有しないことを辺素と呼ぶ。

図16.1

カットの容量

- カット(S, T)においてカットセットに含まれる辺の本数をカット(S, T)の容量と呼び、c(S, T)と表す。

図16.2

最小カット問題

・ s ∈ S、t ∈ Tを満たすカットをs-tカットと呼ぶ。

最小カット問題(辺の容量が1の場合)

・ グラフGから辺を数本取り除いてs-t間を分断する時、
取り除く最小の本数を求めているとも言える。

辺連結度に関する弱双対性

- ある主問題の最小値が、その補問題(双対問題)の最大値以上の値をとるとき、弱双対性が成り立つと呼ぶ。

- 辺連結度については、次の弱双対性が成り立つ

辺連結度に関する問題の弱双対性

辺連結度に関する弱双対性

図16.3

辺連結度に関する強双対性

- ある主問題の最小値が、その補問題(双対問題)の最大値になるとき、強双対性が成り立つと呼ぶ。

- 辺連結度については、次の強双対性が成り立つ

辺連結度に関する問題の強双対性

- この証明及び、実際の辺連結度を求めるアルゴリズムを次のスライド以降で紹介する。

貪欲法に基づく解法

- あるs-tパスを取り、さらに追加でs-tパスを取れるのであれば
  どんどん取っていくという貪欲法が基本的な解法となる。

- しかし明らかに単純な貪欲法ではうまくいかない。下の図から
  追加でs-tパスをとることは出来ない(が、これは最大本数ではない)

図16.4左部

増加パス

- 既に存在するパスを逆流するようなパス(増加パス)は取っても良いものと考えられる。この時、双方向に通過した部分は相殺する。

図16.4右図

残余グラフ

- 増加パスを考えるときは、まず残余グラフを構成すると良い。

- 残余グラフとは、通ったs-tパスの辺の向きを逆方向にしたグラフである。

図16.5

辺連結度を求めるアルゴリズム

- これまでの議論をまとめると、辺連結度を求めるアルゴリズムは次の通りとなる。

2点s-t間の辺連結度を求めるアルゴリズム

強双対性が成り立つ証明

- アルゴリズムの終了時にk本の辺素なs-tパスが得られる。

- これに基づいて容量kのカットを構築することができることを示す。

- すると弱双対性が成立しているため、辺連結性がkであることがわかる。

辺連結性に関する問題の弱双対性

- 同時に、強双対性が成立していることも証明される。

強双対性が成り立つ証明

- アルゴリズム終了時の残余グラフG'を考え、sから到達可能な点をS
  そうではない点をTとする。s-tパスは存在しないので、$s \in S$、$t \in T$
  である。よってこれはs-tカットである。

図16.6

強双対性が成り立つ証明

- 元のグラフ G において S から出ている任意の辺 $e = (u, v)$ ($u \in S$, $v \in T$) は，k 本の s-t パス $P_1, P_2, \ldots, P_k$ のうちのいずれかの中に含まれています (そうでなければ，v も残余グラフ上で頂点 s から到達可能であり，$v \in T$ であることに矛盾します).

- 元のグラフ G において S へと入っている任意の辺 $e = (u, v)$ ($u \in T$, $v \in S$) は，どの s-t パス $P_1, P_2, \ldots, P_k$ の中にも含まれていません (そうでなければ，残余グラフ上では辺 e の向きは逆なので，頂点 u も頂点 s から到達可能ということになり，$u \in T$ であることに矛盾します).

・ よって、u ∈ S、v ∈ Tとなる辺eは各パスと一対一に対応するので、c(S, T) = kである。

・ 頂点sから出る辺の本数は高々|V|-1のため、kはO(|V|)であり、またs-tパスはO(|E|)で求まるため、その計算量はO(|V||E|)となる。

重みつき有向グラフへの拡張

- これまでは、重みなし有向グラフについて取り扱ってきたが、
  重みつき有向グラフでも同様の議論が展開できる。

図16.7

- 重みつきの場合、s-tパスの最大本数ではなく、流せる最大の流量
  (フロー)を計算する(物流などを考える上で明らかに重要)

- これは最大流問題と呼ばれる。(詳細は教科書16.3~16.4節を
  参考のこと)

応用例: 二部マッチング問題

- 二部グラフが与えられたときに、２カテゴリ間で最大何組のペアを作ることができるかという問題。

図16.14

- 商品推薦や従業員シフト割り当てなど応用例は多数

応用例: 二部マッチング問題

- 新たに頂点sとtを追加し、また向きも一方向に定めることで、辺連結度を求める問題と同様の問題となる。

図16.15

まとめ

- 最小全域木問題を解く方法として、貪欲法に基づくクラスカル法を紹介した。

- 重みなし有向グラフにおいて、辺連結度を求めるアルゴリズムを紹介した。この方法は、残余ネットワークを構成しながら貪欲法により解を求めることができる。

- 辺連結度を求めるアルゴリズムは、二部グラフの最大マッチング問題に応用可能である。