Approximation Algorithms

Wei-Yin Lin (林蔚茵)

Department of Computer Science and Information Engineering
National Taiwan University

June 10, 2011

Why approximation algorithms?

如果已知要解的問題是NP-Complete的問題, 該怎麼辦？

- 可找出exact solutions的演算法中，某些problem instances得花上exponential time
- 有些時候不要太執著會比較好
- 想法：多項式時間內可解, 並保證我們所找到的解不會太差
- 想一個快速的方法來找到*near-optimal* solutions（也就是近似演算法）
- 實用性高

Approximation ratio

- 在解一個optimization problem時, 我們希望近似演算法的解和最佳解不會差太多
- 假設input的大小為n, 在最糟的情況下, 近似演算法的解也會被*approximation ratio* $\rho(n)$給限制住
- C：某一個演算法的解
  C^*：最佳解

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

  若達到此ratio, 則稱此演算法為一個$\rho(n)$-*approximation algorithm*
- 此定義在minimization和maximization problems上都適用

Approximation ratio

- 我們希望$\rho(n)$越小越好
- 小到$\rho(n) = 1$時, 最佳解就跑出來了
- ratio的形式不一（等等會看到例子）
  1. a small constant
  2. a function of n
  3. 有些會隨著執行時間的增加而變得越來越好
- 此類的演算法通常都不會很複雜, 但是......

Approximation scheme

Approximation scheme

- Input除了大小為n的problem instance之外, 還多加了一個正的常數ϵ, 使其成為一個(1+ϵ)-approximation algorithm

Polynomial-time approximation scheme (PTAS)

- 任給一個$\epsilon > 0$, 該scheme都可以在多項式時間內執行完畢
- 可以任意地逼近最佳解, 感覺很棒！
- 但是......隨著ϵ的值越小, 所花的時間可能會急遽增加
  比方說時間複雜度為 $O(n^{2/\epsilon})$

Fully polynomial-time approximation scheme (FPTAS)

- 執行時間為n且為$1/\epsilon$的多項式, 例如$O((1/\epsilon)^2 n^3)$
- 如此一來, 隨著ϵ常數倍的減少, 時間必定也只會有常數倍的增加

The vertex-cover problem

Problem 1 (The vertex-cover problem)

A *vertex cover* of an undirected graph $G = (V, E)$ is a subset $V' \subset V$ such that if (u, v) is an edge of G, then either $u \in V'$ or $v \in V'$(or both). The size of a vertex cover is the number of vertices in it. The vertex-cover problem is to find a vertex cover of minimum size in a given undirected graph.

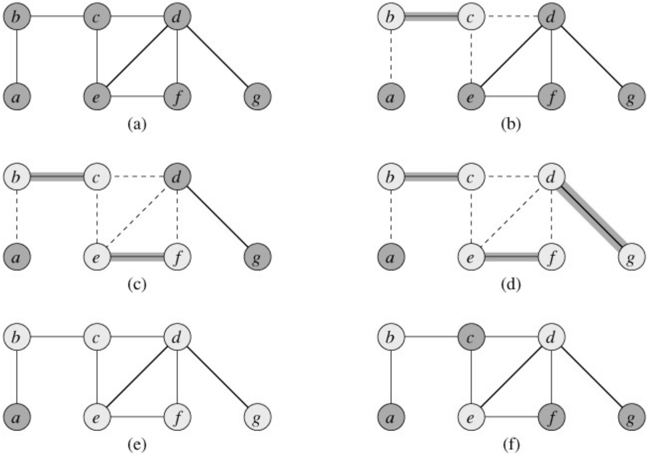
- 此問題的decision版本為NP-Complete (Section 34.5.2)

- 以下的演算法所找到的vertex cover, 其size保證不超過最佳解的兩倍

APPROX-VERTEX-COVER(G)

```
C = ϕ
E' = G.E
while E' ≠ ϕ
    let (u, v) be an arbitrary edge of E'
    C = C ∪ {u, v}
    remove from E' every edge incident on either u or v
return C
```

Example



Theorem

APPROX-VERTEX-COVER is a polynomial-time 2-approximation algorithm

Proof

- 最終回傳的C會是一個vertex cover
- 為什麼C的大小不會超過最佳解大小的兩倍？
  令A為所有在APPROX-VERTEX-COVER的第4行中所挑出的邊
  重要觀察：A中的每個邊皆不會具有共同的端點, 所以
  1. 可得知C^*的lower bound為
     $|C^*| \geq |A|$
  2. 可得知C的(tight) upper bound為
     $|C| = 2|A|$

  $\Rightarrow |C| = 2|A| \leq 2|C^*|$

□

- APPROX-VERTEX-COVER的執行時間為$O(|V|+|E|)$
- 我們不需要知道optimal vertex cover的大小, 就能夠找到approximation ratio！
- 關鍵是甚麼?

The traveling-salesman problem

Problem 2 (The traveling-salesman problem)

Given a complete undirected graph $G = (V, E)$ that has a nonnegative integer cost $c(u, v)$ associated with each edge $(u, v) \in E$, the traveling-salesman problem is to find a hamiltonian cycle (a tour) of G with minimum cost.

- 此問題的decision版本為NP-Complete (Section 34.5.4)

The traveling-salesman problem with the triangle inequality

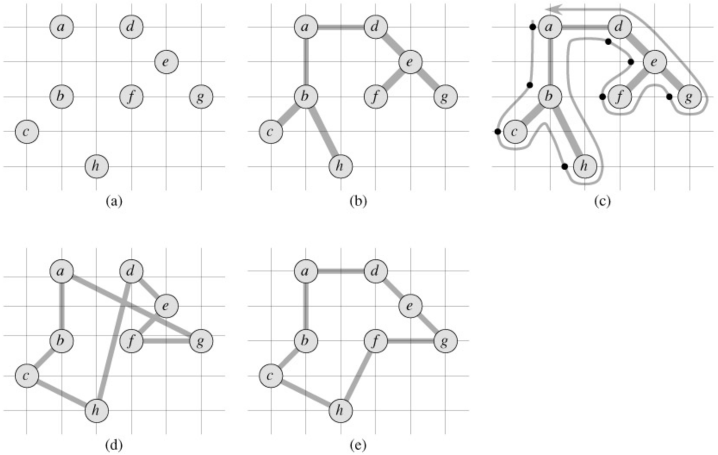
- 令$c(A)$為cost的總合, 其中$A \subseteq E$
  i.e., $c(A) = \sum_{(u,v) \in A} c(u,v)$
- Triangle inequality:
  對所有的$u, v, w \in V$, $c(u,w) \leq c(u,v) + c(v,w)$
- 若給定的cost function符合triangular inequality, 此問題的decision版本依然是NP-Complete (Exercise 35.2-2)
- 但是我們有一個2-approximation algorithm可以來解這個問題
- 想法同the vertex-cover problem, 此時該怎麼找lower bound?

- Input: G: complete undirected graph, c: cost function
- MST-PRIM: 找minimum spanning tree的演算法 (Section 23.2)
- 若c符合三角不等式, 以下演算法可保證找到一個hamiltonian cycle H, 其cost總合不超過optimal hamiltonian cycle的2倍

```
APPROX-TSP-TOUR(G,c)
select a vertex r ∈ G.V to be a "root" vertex
compute a minimum spanning tree T for G from root r
    using MST-PRIM(G, c, r)
let H be a list of vertices, ordered according to when they are first visited
    in a preorder tree walk of T
return the hamiltonian cycle H
```

- 執行時間：$\Theta(|V|^2)$

Example



The cost of H is approximately 19.074. The cost of H^* is approximately 14.715, which is about 23% shorter than H.

Theorem

APPROX-TSP-TOUR is a polynomial-time 2-approximation algorithm for the traveling-salesman problem with the triangle inequality

Proof

- H^*: optimal tour
- W: 在做preorder traversal時, 把過程中第一次和回程時經過的每一點都記錄下來 (稱之為full walk)
  (ex. 承上例, full walk就是 $a, b, c, b, h, b, a, d, e, f, e, g, e, d, a$)
- 將W中前面已出現過的點刪除, 再連回起點, 就是我們要的H
  (ex. 承上例, hamiltonian cycle就是 $a, b, c, h, d, e, f, g, a$)
  1. $c(T) \leq c(H^*)$
  2. $c(W) = 2c(T)$

  $\Rightarrow c(H) \leq c(W) = 2c(T) \leq 2c(H^*)$

□

The general traveling-salesman problem

- 看起來要找個2-approximation algorithm好像不算太難
- 何不來教一個解general TSP的近似演算法呢?
  因為找不到......

Theorem

If P≠NP, then for any constant $\rho \geq 1$, there is no polynomial-time approximation algorithm with approximation ratio ρ for the general traveling-salesman problem.

Proof

- 利用矛盾證法, 假設存在ratio為ρ的近似演算法A可解TSP
  其中$\rho \geq 1$為一常數
- Claim: A可在polynomial time之內解hamiltonian cycle的問題 (HC)
  Proof.
  (a) 將HC的problem instance $G = (V, E)$轉為TSP的problem instance $G' = (V, E')$
    - $E' = \{(u, v) : u, v \in V$ and $u \neq v\}$: 將G轉換成complete graph
      $$c(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E, \\ \rho|V| + 1 & \text{otherwise} \end{cases}$$
    - 此轉換可在polynomial time之內完成

  (b) G中具有一個hamiltonian cycle H
  $\Leftrightarrow (G', c)$具有一個cost為$|V|$的tour
  $\Leftrightarrow A$會回報一個cost不超過$\rho|V|$的tour
  $\because$不在原本G中的邊都太貴了
  i.e., $(\rho|V| + 1) + (|V| - 1) = \rho|V| + |V| > \rho|V|$

$\square$

The set-covering problem

- 到目前為止所看到的都是constant ratio
- 有時候隨著input的大小增加, ratio也會變大
  也就是說, 如果input越大, ouput離最佳解可能會差得越多
- 但如果ratio的成長速度不會太快, 就還可以接受
  比方說, ratio是一個logarithm function
- The set-covering problem主要應用在於解決資源分配的問題
- 此亦為The vertex-cover problem的推廣問題
  $\Rightarrow$ The set-covering problem is NP-hard (Excercise 35.3-2)
  這等等會說明, 先來看看問題的定義吧！

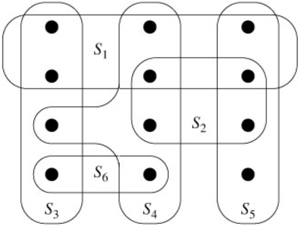
Problem definition

- Problem instance: $(X, \mathcal{F})$
  X: a finite set
  $\mathcal{F}$: a set of subsets of X, where $X = \bigcup_{S \in \mathcal{F}} S$
  X中的每一個元素都會至少屬於$\mathcal{F}$中某一個集合
- $\mathcal{C}$ covers X: $X = \bigcup_{S \in \mathcal{C}} S$

Problem 3 (The set-covering problem)

The set-covering problem is to find a minimum-size subset $\mathcal{C} \subseteq \mathcal{F}$ whose members covers all of X.

Example



- $X = \{$所有的黑點點$\}$
  $\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5, S_6\}$
- A set cover of size 4: $\mathcal{C} = \{S_1, S_4, S_5, S_6\}$
- A minimum size set cover: $\mathcal{C}^* = \{S_3, S_4, S_5\}$

舉個活生生的例子

某國國會要組一個預算審查委員會

- 徵求精通以下項目的人才： 交通, 城市發展, 醫療, 司法, 勞工, 衛生, 教育, 環境保護, 能源, 國土安全, 財政貿易, 外交
- A議員擅長：醫療, 環境保護, 能源
  B議員擅長：城市發展, 財政貿易, 外交
  C議員擅長：交通, 城市發展, 衛生, 教育, 環境保護, 能源, 國土安全, 財政貿易, 外交
  D議員擅長：勞工

  ……
- 為了要節省成本, 委員會的人數不能太多
- C議員超强, 應該要優先考慮將他列為議程委員
  選他就不用選B了
- D很沒用?

A greedy approximation algorithm

```
GREEDY-SET-COVER(X,F)
U = X
C = φ
while U ≠ φ
    select an S ∈ F that maximizes |S ∩ U|
    U = U − S
    C = C ∪ {S}
return C
```

- while迴圈最多會執行$\min(|X|, |\mathcal{F}|)$次
  $\Rightarrow$ 總執行時間為 $O(|X||\mathcal{F}|\min(|X|, |\mathcal{F}|))$
- 可以改進成linear time (Exercise 35.3-3)

GREEDY-SET-COVER的ratio分析

Theorem

GREEDY-SET-COVER is a polynomial-time $\rho(n)$-approximation algorithm, where $\rho(n) = H(\max\{|S| : S \in \mathcal{F}\})$.

在找ratio之前, 先來定義一些我們在證明裡需要東西:

- $H(d) = H_d = \sum_{i=1}^{d} 1/i$
- C^*: 最佳解
  C: 演算法回傳的解
  S_i: 在執行到第i個iteration時所選取的subset

GREEDY-SET-COVER的ratio分析

- 假設每選取完一個subset時, cost就會增加1
- 將執行完第i輪時的cost均分給所有在S_i才第一次被cover到的element
- 也就是說, 令c_x為x被分到的cost值, 則 $c_x = \frac{1}{|S_i - (S_1 \cup S_2 \cup \ldots \cup S_{i-1})|}$

Claim:

For any set S belonging to the family $\mathcal{F}$, $\sum_{x \in S} c_x \leq H(|S|)$.

(此不等式的證明請參考P1120-1121)

GREEDY-SET-COVER的ratio分析

Proof

$$
\begin{aligned}
|\mathcal{C}| &= \sum_{x \in X} c_x && (1)\\
&\leq \sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x && (2)\\
&\leq \sum_{S \in \mathcal{C}^*} H(|S|) && (3)\\
&\leq |\mathcal{C}^*| \cdot H(\max\{|S| : S \in \mathcal{F}\}). && (4)
\end{aligned}
$$

□

(3): according to the above claim

GREEDY-SET-COVER的ratio分析

- 由上述的定理與不等式$H_n \leq \ln n + 1$可得到以下的推廣：

Corollary

GREEDY-SET-COVER is a polynomial-time $(\ln |X| + 1)$-approximation algorithm

GREEDY-SET-COVER的ratio分析

- 在某些特定的情形下, $\max\{|S| : S \in \mathcal{F}\}$會是一個不太大的常數
- 此時在應用上反而可以得到一個還不錯的ratio
- 譬如說在一個degree不超過3的圖上找vertex cover, 利用GREEDY-SET-COVER, 我們會得到一個$\rho(n) = \frac{11}{6}$的近似演算法
- 這比之前提的GREEDY-SET-COVER還要好

The vertex-cover problem $\leq_p$ The set-covering problem

- 假設$G = (V, E)$為The vertex-cover problem的problem instance
- 問題的轉換方式如下：
  $X = E$, $n = |V|$
  $\mathcal{F} = \{S_1, S_2, ..., S_n\}$
  $S_i = \{e_j \mid e_j \text{ is incident with } v_i\}$, for $i = 1$ to n
- G中具有size為k的vertex cover
  $\Leftrightarrow$ $(X, \mathcal{F})$具有size為k的set cover

永無止盡地追求更好的ratio...

- 若限制input G中所有的點的degree皆不超過3
- 執行GREEDY-SET-COVER, 最終可得一vertex cover H, 其大小不超過最佳解的$H(3) = 11/6$倍
- $2 \rightarrow 15/8 \rightarrow 3/2 \rightarrow$ PTAS $\rightarrow ?$

超感謝大家的啦！ :D