Lab09-Approximation Algorithm

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Jiahao Fan.

∗ Name:田雪飞 Student ID:515030910347 Email: 13487426939@qq.com

1. **Solution.** The following is process of solution.

a. The basic idea of a greedy algorithm is starting with an arbitrary center, and in each round, add the 'farthest'vertex to the center set until there are totaly k centers.

Algorithm 1: Greedy approximation.

Input: an complete undirected graph with nonnegative edge G(V,E).
Output: minimal $max_v\{cost(v, S)\}$.

S $\leftarrow \{c_1\}$; c_1 is an arbitrary center we choose at first .
for $\boldsymbol{v} \in$ ***V-S*** **do**
  if $|S| \le k$ **then**
    Find the farthest vertex v_i to center set.
    S $\leftarrow$ **S** $\cup \{c_i\}$.
return ***S***.

b. Let's consider a set **S** and two vertices v_g and v_opt , The optimal choice is to choose v_{opt} but we choose the v_g by greedy algorithm. In this case, we can use an illustration to describe this question, then we can construct the following relation:

$$\begin{cases} cost(v_g, S) = max\{s_1, s\} = s_{m1}; \\ cost(v_{opt}, S) = mas\{s_2, s\} = s_{m2}; \end{cases}$$

According to the triangle inequality:

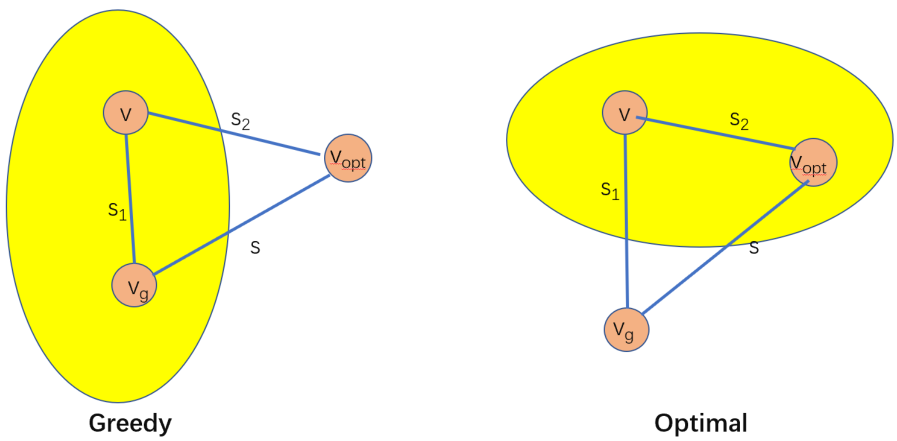

Figure 1: The flow of two nodes is 1

$cost(v_g, S) = max\{s_1, s\} = s_{m1} \le s_2 + s \le 2 * max\{s_2, s\} = 2cost(v_{opt}, S)$;
then $cost(v_g, S) \le 2cost(v_{opt}, S)$,so $\frac{cost(v_g,S)}{cost(v_{opt},S)} \le 2$.

□

2. **Solution.** The following is the process of proof.
proof: In this case, we can use a vertex **v** which belongs to **R** , set **R-v** and set **S**, the goal is to find minimum cost tree, we can use an illustration to describe this question:

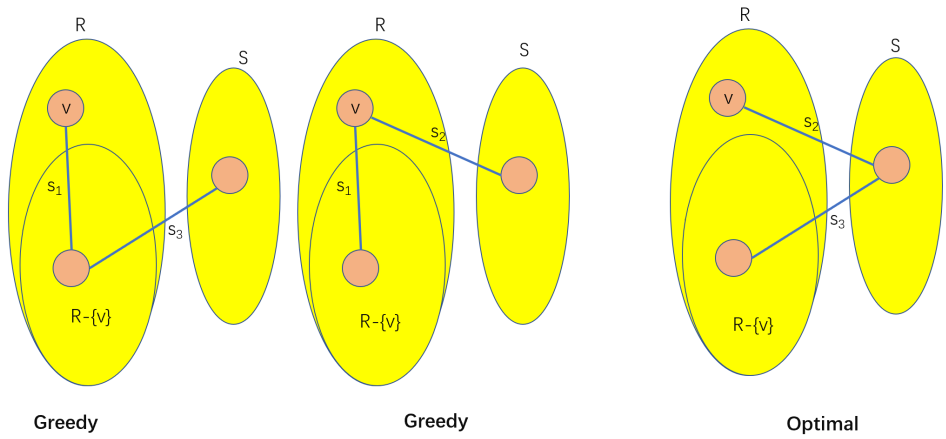

Figure 2: The flow of two nodes is 1

we consider the a vertex **v** in **R**,according to approximation, we know vertex **v** must connect to one vertex in **R-v**, but in optimal answer,we know **v** can connect to one vertex in **S**, so, we can construct to the following relation by illustration above.

$$\begin{cases} c_g = s_1 + min\{s_2, s_3\}; \\ c_{opt} = s_2 + s_3; \end{cases}$$

According to the triangle inequality: $c_g = s_1 + min\{s_2, s_3\} \leq 2(s_2+s_3) \leq 2c_{opt}$; For all vertices in R,we have same relation,so $C_g \leq 2C_{opt}$, then $\frac{C_g}{C_{opt}} \leq 2$.

□

3. **Minimum Weighted Vertex Cover:**
a. Denote: we assume C is a cover vertex set, In graph G(V,E), for each vertex v ∈ V, if v ∈ C, $x(v) = 1$ (vertex v in vertex cover set), if v ∉ C, $x(v) = 0$ (vertex v not in vertex cover set).at least one vertex is in vertex cover set for every edge:$x(v_j) + x(v_i) \geq 1$;
integer linear program:

$$\begin{cases} \sum_{v_i \in V} c_i x(v_i); \\ x(v_j) + x(v_i) \geq 1; \\ x(v_i), (v_j) = 0, 1; \\ i \in \{1, 2 \ldots n\} \end{cases}$$

b. proof: As we know, the following is an approximation algorithm with value $m_{LP}(G)$, we assume: m^*LP is the best solution for linear program. Then $m^*(LP) \leq m^*(G)$;
In solution **a**, we know $x(v_j) + x(v_i) \geq 1$;
so there must have one vertex which $x(v) \geq \frac{1}{2}$; $\{v_i, v_j\} \geq \frac{1}{2}$;
$m^*(LP) = \sum_{v_i \in V} c_i x(v_i) \geq \sum_{v_i \in V and x(v) \geq \frac{1}{2}} c_i x(v_i) \geq \sum_{v_i \in V and x(v) \geq \frac{1}{2}} \frac{1}{2} * c_i = \frac{1}{2} m_{LP}(G)$
Then, $\Rightarrow \frac{1}{2} m_{LP}(G) \leq m^*(G)$;
So ,$\Rightarrow m_{LP}(G)/m^*(G) \leq 2$.

4. Give the corresponding $(I, sol, m, goal)$ for **Metric k-center** and **Minimum Weighted Vertex Cover** respectively. **Metric k-center**(I,sol,m,goal) :
   I={G(V,E)|G is a graph}; poly time decidable.
   sol={U $\subseteq V|$arbitrary select a vertex as center, add the 'farthest' vertex to the center set until there are totaly k centers.}
   m=farthest distance v to k-center set U; $v \not inU$;
   $goal = min\{m\}$;
   Minimum Weighted Vertex Cover (I,sol,m,goal):
   I={G(V,E)|G is a graph};poly time decidable.
   $sol = U \subseteq V|\forall(v,u) \in E(v \in V or u \in V)$.
   $m = \sum_{v \in V} c(v)x(v).v \in V, x(v) = 1; v \notin V, x(v) = 0$;c(v) is the cost of vertex v.
   $goal = min\{m\}$

Remark: You need to include your .pdf and .tex files in your uploaded .zip file.