Chapter 2
CDS in General Graph

Leadership is based on inspiration,
not domination; on cooperation, not intimidation.
WILLIAM ATHUR WARD

2.1 Motivation and Overview

Since MIN-CDS is NP-hard, approximation algorithm design becomes an important issue in study of CDS. What is the complexity of approximation for MIN-CDS? Guha and Khuller [62] showed that MIN-CDS has no polynomial-time $(\rho \ln n)$-approximation for $0 < \rho < 1$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$ where n is the number of vertices in input graph. Moreover, they designed a 2-stage greedy algorithm with performance ratio $3 + \ln\delta$ where δ is the maximum vertex degree of input graph. The effort on improvement of this 2-stage greedy algorithm encounted an essential difficulty on analysis of greed approximation.

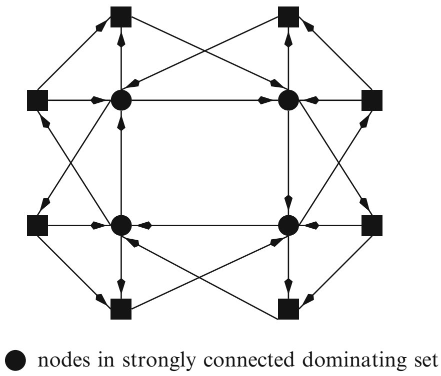
In 1982, Wolsey [116] discovered a general theorem on analysis of greedy approximation with submodular potential functions, which covers many existing results. For example, the greedy algorithm for WCDS in [17] has a submodular potential function and can be analyzed with Wolsey Theorem. Since Wolsey Theorem was established, the submodularity becomes an important property for algorithm designer to seek. Unfortunately, the potential function used in Guha–Khuller's Greedy Algorithm is not submodular, and so far, no one has found a submodular potential function to design a greedy approximation for MIN-CDS.

How do we analyze the greedy approximation with a nonsubmodular potential function? Ruan et al. [92] found a technique and designed a one-stage greedy approximation for MIN-CDS with performance ratio $2 + \ln\delta$. Du et al. [40] found more techniques and designed a greedy approximation scheme for MIN-CDS with performance ratio $a(1 + \ln\delta)$ for any $a > 1$.

However, those techniques do not work in weighted version of MIN-CDS.

D.-Z. Du and P.-J. Wan, *Connected Dominating Set; Theory and Applications*, Springer Optimization and Its Applications 77, DOI 10.1007/978-1-4614-5242-3_2,

Fig. 2.1 Strongly connected dominating set



MINW-CDS: Given a connected graph $G=(V,E)$ with vertex weight $w: V \to R^+$, find a CDS with minimum total weight.

For MINW-CDS, Guha and Khuller [62] proposed a two-stage approximation algorithm as follows: At the first stage, construct a dominating set D with a greedy approximation for MINW-SET-COVER in which each node v corresponds to the subset of nodes dominated by the node v.

MINW-SET-COVER: Given a collection $\mathcal{C}$ of subsets of X with a nonnegative weight function $w: \mathcal{C} \to R^+$, find a set cover with minimum total weight.

At the second stage, connect D into a CDS with a greed approximation, given by Klein and Ravi [69], for node-weighted Steiner tree problem.

NODE-WEIGHTED STEINER TREE: Given a graph $G=(V,E)$ with nonnegative node weight $w: V \to R^+$, and a subset P of nodes, find a subset S of nodes with minimum total weight, interconnecting all nodes in P.

Note that the total weight of dominating D can be upper-bounded by $(1+\ln n)\cdot$ opt and the total weight of Steiner nodes S added in the second stage can be upper-bounded by $(1+2\ln 2)\cdot$ opt where opt is the minimum weight of a CDS. Therefore, the approximation given by Guha and Khuller [62] has performance ratio at most $2+3\ln n$. Soon later, Guha and Khuller [63] found that the technique initiated by Klein and Ravi [69] can be directly employed to design greedy approximations for MIN WEIGHT CDS. Actually, this technique works in a wide range of area. The disadvantage of this technique is that obtained performance ratio is a little large. Guha and Khuller [63] also improved this technique. Using improved technique, they designed a polynomial-time approximation for MINW-CDS with performance ratio approaching $1.35\cdot\ln\delta$, which is the best known so far.

Consider a directed graph $G=(V,E)$. Thai and Du [101] and Li et al. [74] introduce the concept of CDS into directed graphs. A node subset C is a *dominating set* if every node not in C has an arc going to C and an arc coming from C. Furthermore, C is called a *strongly connected dominating set (SCDS)* if subgraph induced by C is strongly connected (Fig. 2.1). They study the following problem.

MIN-SCDS: Given a directed graph, find a strongly connected dominating set with minimum cardinality.

Li et al. [76, 77] found a construction of SCDS by using the solution for MINW-BROADCAST and hence obtained polynomial-time $(2+4\ln n)$-approximation [76] and $(2+3\ln n)$-approximation [77] for MIN-SCDS, respectively.

2.2 Complexity of Approximation

Consider the following problem.

MIN-SET-COVER: Given a collection $\mathcal{C}$ of subsets of a finite set X, find a set cover from $\mathcal{C}$, with minimum cardinality.

MIN-SET-COVER has the following inapproximability.

Theorem 2.2.1 (Feige [47]). *For* $0 < \rho < 1$, *there is no polynomial-time* $(\rho \ln n)$-*approximation for* MIN-SET-COVER *unless* $NP \subseteq DTIME(n^{O(\log\log n)})$.

Using this result, Guha and Khuller [62] established the inapproximability of MIN-CDS.

Theorem 2.2.2 (Guha and Khuller [62]). *For* $0 < \rho < 1$, MIN-CDS *has no polynomial-time* $(\rho \ln n)$*-approximation unless* $NP \subseteq DTIME(n^{O(\log\log n)})$ *where* n *is the number of vertices in input graph.*

Proof. We recall the reduction from SET-COVER to MIN-CDS in the proof of NP-hardness of MIN-CDS in Theorem 1.1.3. The reduction can also be seen as a reduction from MIN-SET-COVER to MIN-CDS as follows.

For any instance of MIN-SET-COVER, a collection $\mathcal{C}$ of m subsets of a set X of n elements, the reduction constructs a graph $G = (V, E)$ with vertex set

$$V = X \cup \mathcal{C} \cup \{s, t\}$$

and edge set

$$E = \{(x, S) \mid x \in S \text{ for } x \in X, S \in \mathcal{C}\} \cup \{(s, S) \mid S \in \mathcal{C}\} \cup \{(s, t)\}.$$

This reduction has been proved to have property that $\mathcal{C}$ has a set cover of size at most k if and only if G has a CDS of size at most $k+1$. Consequently, the minimum set cover of $\mathcal{C}$ contains k subsets if and only if the minimum CDS of G contains $k+1$ vertices.

Now, suppose for some $0 < \rho < 1$, there is a polynomial-time $(\rho \ln n)$-approximation for MIN-CDS. We prove $NP \subseteq DTIME(n^{O(\log\log n)})$.

Choose a positive integer $k_0 > \frac{\rho}{1-\rho}$. Then $\rho(1 + \frac{1}{k_0}) < 1$. Choose a positive number ρ' such that $\rho(1 + \frac{1}{k_0}) < \rho' < 1$. We then show that MIN-SET-COVER in the special case $|\mathcal{C}| = m \leq n$ also has a polynomial-time approximation with performance ratio $\rho' \ln n$.

For each input collection $\mathcal{C}$ in MIN-SET-COVER, we first check all subcollections of at most k_0 subsets whether it is a set cover or not. This takes time bounded by a polynomial of degree k_0.

If no set cover of cardinality k_0 is found, then any set cover of $\mathcal{C}$ contains at least $k_0 + 1$ subsets.

Suppose C is $(\rho \ln n)$-approximation solution for MIN-CDS. Then C has size at most $(\rho \ln(m+n+2))(k+1)$. Thus, we can obtain a set cover $\mathcal{A}$ of size at most $\rho \ln(m+n+2)(k+1) - 1 < \rho(1+\frac{1}{k_0})(1+\frac{\ln 3}{\ln n})(\ln n)k$ where $\mathcal{A} = \mathcal{C} \cap C$. When n is sufficiently large, $\mathcal{A}$ is a $\rho' \ln n$-approximation solution for MIN-SET-COVER. By Theorem 2.2.1, $NP \subseteq DTIME(n^{O(\log\log n)})$. □

There is another lower bound result for MIN-SET-COVER.

Theorem 2.2.3 (Raz and Safra [89]). *There is a constant $c > 0$ such that the existence of polynomial-time $(c \ln n)$-approximation for* MIN-SET-COVER *implies $NP = P$.*

Following from this result, we can also obtain a similar result for MIN-CDS.

Theorem 2.2.4. *There is a constant $c > 0$ such that the existence of polynomial-time $(c \ln n)$-approximation for* MIN-CDS *implies $NP = P$.*

2.3 Two-Stage Greedy Approximation

Consider a graph G and a subset C of vertices in G. We divide all vertices in G into three classes with respect to C:

- *Black* vertices: vertices in C.
- *Grey* vertices: vertices not in C but dominated by black vertices.
- *White* vertices: vertices not dominated by black vertices.

Clearly, C is a CDS if and only if there does not exist a white vertex and the subgraph induced by black vertices is connected. Let $p(C)$ be the number of connected components of $G[C]$, the subgraph of G induced by C, and $h(C)$ the number of white vertices. Let $g(C) = p(C) + h(C)$. Then C is a CDS if and only if $g(C) = 1$. We may use g to design an algorithm as follows:

Greedy Algorithm GK:
input a connected graph G.
Set $C \leftarrow \emptyset$;
while there exists a vertex x such that $g(C \cup \{x\}) < g(C)$ **do**
    choose a vertex x to minimize $g(C \cup \{x\})$ and
    set $C \leftarrow C \cup \{x\}$;
output C.

However, this algorithm may not output a CDS. Indeed, even if for vertex x, $g(C \cup \{x\}) = g(C)$, C may not be a CDS. An example is shown in Fig. 2.2. In fact,

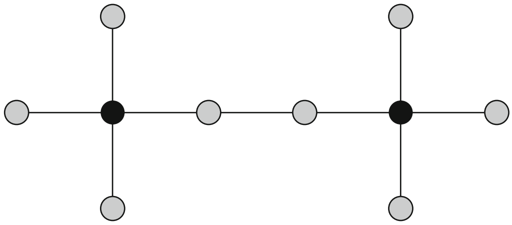


Fig. 2.2 Output of Algorithm GK may not be a CDS

what appeared in this example is a typic case. If a white vertex exists, then let x be a gray vertex adjacent to a white vertex, then we must have $g(C \cup \{x\}) < g(C)$. Therefore, for C obtained from **Greedy Algorithm GK**, no white vertex exists. This means that if output C is not a CDS, then C does not induced a connected subgraph. In such a case, its connected components are apart not very far. Since the given graph is connected, all black components are connected together through some chains of two adjacent gray vertices. To see this, we first note that no gray vertex is adjacent to two black components since coloring such a gray vertex in black would reduce the value of potential function. Now, for contradiction, suppose that all black components cannot be connected through chains of two adjacent gray vertices. Then, we can divide all black vertices into two parts such that the distance between the two parts is more than three, say $k > 3$. Consider the path between the two parts, $(u, x_1, x_2, \ldots, x_{k-1}, v)$, which reaches the distance between the two parts, that is, u and v belong to the two parts respectively, $x_1, x_2, \ldots, x_{k-1}$ are gray vertices with $k-1 \geq 3$, and no shorter path of this type exists. Since x_2 is grey, it must be adjacent to a black vertex w. If w and u are in the same part, then the path from w to v indicates that the distance between the two parts is at most $k-1$, a contradiction. If w and v are in the same part, then the path from u to w indicates that the distance between the two parts is at most $3 < k$, also a contradiction.

Based on above observations, Guha and Khuller [62] designed a two-stage greedy algorithm as follows.

Guha–Khuller Algorithm:
input a connected graph G.
Stage 1
  Employ **Greedy Algorithm GK** to obtain a dominating set C;
Stage 2
  while there are more than one black components **do**
    find a chain of two gray vertices x and y connecting at least
    two black components and $C \leftarrow C \cup \{x, y\}$;
output C.

In this two-stage greedy approximation, stage 1 is a greedy algorithm computing a dominating set and stage 2 connects this dominating set into a connected one. In the potential function $g(C)$, $h(C)$ is used for issuing that Stage 1 gives a dominating set, and $p(C)$ is used for making the number of black connected components smaller.

Theorem 2.3.1 (Guha and Khuller [62]). *Suppose input graph is not a star. Then, Guha–Khuller Algorithm is a polynomial-time* $(3+\ln\delta)$*-approximation for CDS where* δ *is the maximum vertex degree of input graph.*

Proof. By a piece, we mean a white vertex or connected component of subgraph induced by black vertices. A piece is said to be *touched* by a vertex x if x is in the piece or adjacent to the piece. For any vertex subset C, the number of piece is $g(C)$. Suppose $x_1,\ldots,x_g$ are selected in turn by Guha–Khuller Algorithm at stage 1. Denote $C_i=\{x_1,\ldots,x_i\}$ for $1\le i\le g$ and $C_0=\emptyset$. Then, each vertex x touches $1+g(C_{i-1})-g(C_{i-1}\cup\{x\})$ pieces with respect to C_{i-1} and x_i reaches the maximum of this number. Suppose opt is the number of vertices in a minimum CDS. Since a dominating set must touch all pieces, there exists a vertex touches at least $\lceil g(C_{i-1})/\text{opt}\rceil$ pieces. Therefore

$$1+g(C_{i-1})-g(C_i)\ge\frac{g(C_{i-1})}{\text{opt}}$$

that is,

$$g(C_i)\le g(C_{i-1})\left(1-\frac{1}{\text{opt}}\right)+1.$$

Set $a_i=g(C_i)-\text{opt}$. Then,

$$a_i\le a_{i-1}\left(1-\frac{1}{\text{opt}}\right).$$

Clearly, as long as $a_{i-1}>0$, we have $a_i<a_{i-1}$. Therefore, we must have $a_g\le 0$. Choose $j\le g$ such that $a_j\le 0<a_{j-1}$. Then $a_g\le j-g$. This means that when stage 1 ends, at most $\text{opt}-(g-j)+1$ pieces exist and hence at most $\text{opt}-(g-j)+1$ connected black components exist. Therefore, at most $2(\text{opt}-g+j)$ vertices would be added in stage 2. Choose i such that $a_{i+1}<\text{opt}\le a_i$. Then, $j-i\le\text{opt}$ and

$$\text{opt}\le a_{i-1}\left(1-\frac{1}{\text{opt}}\right)\le a_0\left(1-\frac{1}{\text{opt}}\right)^i\le n\mathrm{e}^{-\mathrm{i}/\text{opt}},$$

where n is the number of vertices of input graph. Thus,

$$i\le\text{opt}\ln(n/\text{opt}).$$

Therefore,

$$g+2(\text{opt}-g+j)\le 2\text{opt}+j\le 3\text{opt}+i\le\text{opt}(3+\ln(n/\text{opt}))\le\text{opt}(3+\ln\delta)$$

for $\text{opt}\ge 2$. □

2.4 Weakly CDS

Improving Guha and Khuller's Greedy Algorithm is not an easy job. Actually, this encounted a fundamental difficulty on analysis of greedy algorithms. To explain this, let us use WCDS as an example to introduce the theory of submodular function.

Consider a graph $G = (V,E)$. For any vertex subset C, denote by $q(C)$ the number of connected components of the subgraph with vertex set V and the edge set consisting of all edges incident to vertices in C.

A dominating set C is called a *weakly CDS (WCDS)* if $q(C) = 1$. Chen and Liestman [17] studied the following problem.

MIN-WCDS: Given a graph G, find a WCDS with the minimum cardinality.

They designed a greedy algorithm with potential function $q(C)$.

Chen–Liestman Algorithm
input graph $G = (V,E)$.
$C \leftarrow \emptyset$;
while $q(C) \geq 2$ **do**
  choose $u \in V$ to mimimize $q(C \cup \{u\})$
  $C \leftarrow C \cup \{u\}$;
output C.

The performance ratio of this algorithm is guaranteed by the following theorem.

Theorem 2.4.1 (Chen and Liestman [17]). *Chen–Liestman Algorithm produces an approximation solution within a factor of* $(1+\ln\delta)$ *from optimal, where* δ *is the maximum vertex degree of input graph.*

Actually, the above result has been covered by a general theory on submodular function proposed by Wolsey [116].

Consider a finite set X and a real function f defined on 2^X, the collection of all subsets of X. f is *submodular* if for any two subsets A and B of X,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B).$$

f is *increasing* if for $A \subset B$, $f(A) \leq f(B)$. The *marginal value* of B with respect to A is defined by

$$\Delta_B f(A) = f(A \cup B) - f(A).$$

When $B = \{x\}$ for some $x \in X$, we simply write $\Delta_x f(A)$ instead of $\Delta_{\{x\}} f(A)$ for the marginal value of $\{x\}$ (or simply the marginal value of x) with respect to A. Both monotonicity and submodularity of a function f can be characterized in terms of the marginal values [4,52,84,116].

Lemma 2.4.2. *f is submodular and increasing if and only if for any $x \in X$,*

$$A \subset B \Rightarrow \Delta_x f(A) \geq \Delta_x f(B).$$

Wolsey [116] studied the following problem and greedy algorithm.

MIN-SUBMODULAR-COVER: Given a submodular and increasing function $f : 2^X \to R$ and a nonnegative cost function $c : X \to R^+$, find $A \subseteq X$ to minimize $C(A) = \sum_{x \in A} c(x)$ under constraint $f(A) = f(X)$.

Wolsey Greedy Algorithm
input a monotone increasing submodular function $f : 2^X \to R$;
Initially, set $A \leftarrow \emptyset$;
while $f(A) < f(X)$ **do**
  choose $x \in X - A$ to maximize $\frac{\Delta_x f(A)}{c(x)}$
  $A \leftarrow A \cup \{x\}$;
output A.

The performance of this algorithm is guaranteed by the following theorem [116].

Theorem 2.4.3 (Wolsey Theorem). *Suppose f is a submodular, monotone increasing integer function on 2^X with $f(\emptyset) = 0$. Then, Wolsey Greedy Algorithm produces a $H(\gamma)$-approximation for* MIN-SUBMODULAR-COVER *where $\gamma = \max_{x \in X} f(\{x\})$.*

Proof. Let $x_1, x_2, \ldots, x_k$ be the sequence of elements selected by Wolsey Greedy Algorithm and $A = \{x_1, x_2, \ldots, x_k\}$. Let A^* be an optimal solution of MIN-SUBMODULAR-COVER. We prove

$$c(A) \leq H(\gamma) c(A^*)$$

by a charging argument. Denote $A_0 = \emptyset$ and $A_i = \{x_1, x_2, \ldots, x_i\}$ for each $1 \leq i \leq k$. Denote $\mu_0 = 0$ and $\mu_i = \frac{c(x_i)}{\Delta_{x_i} f(A_{i-1})}$ for each $1 \leq i \leq k$. The parameter μ_i is the referred to as the average price per increment of coverage by x_i for each $1 \leq i \leq k$. We claim that

$$\mu_0 \leq \mu_1 \leq \mu_2 \leq \cdots \leq \mu_k.$$

Indeed, the first inequality is trivial. For any $1 \leq i < k$,

$$\mu_i = \frac{c(x_i)}{\Delta_{x_i} f(A_{i-1})} \leq \frac{c(x_{i+1})}{\Delta_{x_{i+1}} f(A_{i-1})} \leq \frac{c(x_{i+1})}{\Delta_{x_{i+1}} f(A_i)} = \mu_{i+1},$$

where the first inequality follows from the greedy rule and the second inequality follows from the submodularity of f. Thus, our claim holds. Now for iteration i with $1 \leq i \leq k$, we charge each $e \in A^*$ with $\mu_i (\Delta_e f(A_{i-1}) - \Delta_e f(A_i))$. Then, the total charge on each $e \in A^*$ is

$$\sum_{i=1}^{k} \mu_i (\Delta_e f(A_{i-1}) - \Delta_e f(A_i)),$$

and the total charge on A^* is

$$\sum_{e\in S}\sum_{i=1}^{k}\mu_i\left(\Delta_e f\left(A_{i-1}\right)-\Delta_e f\left(A_i\right)\right).$$

We claim that

1. $\sum_{i=1}^{k} c\left(x_i\right)$ is no more than the total charge on A^*.
2. The total charge on $e\in A^*$ is at most $H\left(\gamma\right)c\left(e\right)$.

The first claim is true because

$$\begin{aligned}
\sum_{i=1}^{k} c\left(x_i\right) &= \sum_{i=1}^{k}\mu_i\Delta_{x_i} f\left(A_{i-1}\right)\\
&= \sum_{i=1}^{k}\mu_i\left(f\left(A_i\right)-f\left(A_{i-1}\right)\right)\\
&= \sum_{i=1}^{k}\mu_i\left(\left(f\left(A^*\right)-f\left(A_{i-1}\right)\right)-\left(f\left(A^*\right)-f\left(A_i\right)\right)\right)\\
&= \sum_{i=1}^{k}\left(\mu_i-\mu_{i-1}\right)\left(f\left(A^*\right)-f\left(A_{i-1}\right)\right)\\
&\le \sum_{i=1}^{k}\left(\mu_i-\mu_{i-1}\right)\sum_{e\in S}\Delta_e f\left(A_{i-1}\right)\\
&= \sum_{e\in S}\sum_{i=1}^{k}\left(\mu_i-\mu_{i-1}\right)\Delta_e f\left(A_{i-1}\right)\\
&= \sum_{e\in S}\sum_{i=1}^{k}\mu_i\left(\Delta_e f\left(A_{i-1}\right)-\Delta_e f\left(A_i\right)\right).
\end{aligned}$$

Next, we prove the second claim. Consider an arbitrary element $e\in A^*$. Let l be the first i such that $\Delta_e f\left(A_i\right)=0$. For each $1\le i\le l$, by the greedy rule,

$$\mu_i=\frac{c\left(x_i\right)}{\Delta_{x_i} f\left(A_{i-1}\right)}\le\frac{c\left(e\right)}{\Delta_e f\left(A_{i-1}\right)}.$$

Hence,

$$\begin{aligned}
&\sum_{i=1}^{k}\mu_i\left(\Delta_e f\left(A_{i-1}\right)-\Delta_e f\left(A_i\right)\right)\\
&\quad=\sum_{i=1}^{l-1}\mu_i\left(\Delta_e f\left(A_{i-1}\right)-\Delta_e f\left(A_i\right)\right)+\mu_l\Delta_e f\left(A_{l-1}\right)
\end{aligned}$$

$$\leq c(e)\left(1+\sum_{i=1}^{l-1}\frac{\Delta_e f(A_{i-1})-\Delta_e f(A_i)}{\Delta_e f(A_{i-1})}\right)$$
$$\leq c(e)\left(1+\sum_{i=1}^{l-1}\left(H(\Delta_e f(A_{i-1}))-H(\Delta_e f(A_i))\right)\right)$$
$$= c(e)\left(1+H(\Delta_e f(\emptyset))-H(\Delta_e f(A_{l-1}))\right)$$
$$\leq c(e)\left(1+H(\gamma)-H(1)\right)$$
$$= c(e)H(\gamma).$$

So, the second claim also holds.

The two claims imply that

$$\sum_{i=1}^{k} c(x_i) \leq H(\gamma)\sum_{e\in S} c(e). \qquad \square$$

Now, we return to MIN-WCDS and note the following.

Lemma 2.4.4. $|V|-q(A)$ *is a submodular, monotone increasing integer function with* $|V|-q(\emptyset)=0$.

Proof. Note that q is an integer function and $q(\emptyset)$. By Lemma 2.4.2, it suffices to show that for any $v \in V$,

$$A \subset B \Rightarrow \Delta_v q(A) \leq \Delta_v q(B).$$

Let $H(A)$ denote the graph with vertex set V and edge set consisting of all edges incident to a vertex in A. Then, each connected component of graph $H(B)$ is constituted by one or more connected components of graph $H(A)$. Thus, the number of connected components of $H(B)$ adjacent to v is no more than the number of connected components of $H(A)$ adjacent to v. Therefore, the lemma holds. $\square$

If we set $f(A)=|V|-q(A)$, then it is easy to see that MIN-SUBMODULAR-COVER becomes MIN-WCDS, Wolsey Greedy Algorithm becomes Chen–Liestman Algorithm, and Theorem 2.4.1 can result from Wolsey Theorem.

2.5 One-Stage Greedy Approximation

When a potential function is not submodular, how do we analyze a greedy algorithm with it? We study this problem in this section.

To avoid the above counterexample, we replace $h(C)$ by $q(C)$ the number of connected components of the subgraph with vertex set V and edge set $D(C)$, where $D(C)$ be the set of all edges incident to vertices in C. Define $f(C)=p(C)+q(C)$.

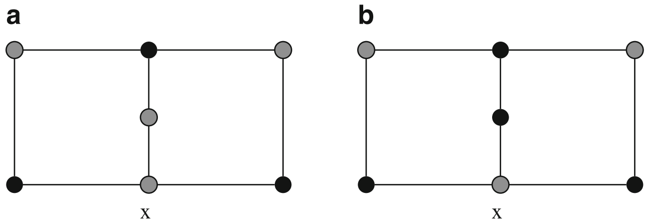


Fig. 2.3 A counterexample ($\Delta_x f(A) > \Delta_x f(B)$ but $A \subset B$)

Lemma 2.5.1. *Suppose G is a connected graph with at least three vertices. Then, C is a CDS if and only if $f(C \cup \{x\}) = f(C)$ for every $x \in V$.*

Proof. If C is a CDS, then $f(C) = 2$, which reaches the minimum value. Therefore, $f(C \cup \{x\}) = f(C)$ for every $x \in V$.

Conversely, suppose $f(C \cup \{x\}) = f(C)$ for every $x \in V$. First, C cannot be the empty set. In fact, for contradiction, suppose $C = \emptyset$. Since G is a connected graph with at least three vertices, there must exist a vertex x with degree at least two and for such a vertex x, $f(C \cup \{x\}) < f(C)$, a contradiction. Now, we may assume $C \neq \emptyset$. Consider a connected component of the subgraph induced by C. Let B denote its vertex set which is a subset of C. For every gray vertex y adjacent to B, if y is adjacent to a white vertex or a gray vertex not adjacent to B, then we must have $p(C \cup \{y\}) < p(C)$ and $q(C \cup \{y\}) \leq q(C)$; if y is adjacent to a black vertex not in B, then $p(C \cup \{y\}) \leq p(C)$ and $q(C \cup \{y\}) < q(C)$; hence, in all cases $f(C \cup \{y\}) < f(C)$, a contradiction. Therefore, every gray vertex adjacent to B cannot be adjacent to any vertex neither in B nor adjacent to B. Since G is connected, it follows that every vertex of G must belong to B or adjacent to B. That is, $B = C$ is a CDS. □

This lemma means that with $-f$ as potential function, Wolsey Greedy Algorithm would produce a CDS. If f is a monotone decreasing, submodular function, then we could directly employ Wolsey Theorem to give an estimation on performance ratio of the algorithm. Unfortunately, f is not submodular. A counterexample is shown in Fig. 2.3.

Could we also give analysis of Wolsey Greedy Algorithm in this case? The answer is yes and a new technique can be introduced based on two observations in the following.

The first observation is that MIN-CDS is an unweighted problem and in unweighted case, there is a simpler analysis for Wolsey Greedy Algorithm.

A Simple Analysis of Wolsey Greedy Algorithm in Unweighted Case: Let $x_1, \ldots, x_g$ be subsets selected in turn by Wolsey Greedy Algorithm. Denote $A_i = \{x_1, \ldots, x_i\}$. Let opt be the number of subsets in a minimum submodular cover. Let $C = \{y_1, \ldots, y_{\text{opt}}\}$ be a minimum submodular cover. Denote $C_j = \{y_1, \ldots, y_j\}$.

By the greedy rule,

$$f(A_{i+1}) - f(A_i) = \Delta_{x_{i+1}} f(A_i) \geq \Delta_{y_j} f(A_i)$$

for $1 \leq j \leq \text{opt}$. Therefore,

$$f(A_{i+1}) - f(A_i) \geq \frac{\sum_{j=1}^{\text{opt}} \Delta_{y_j} f(A_i)}{\text{opt}}.$$

On the other hand,

$$\begin{aligned}\frac{f(C) - f(A_i)}{\text{opt}} &= \frac{f(A_i \cup C) - f(A_i)}{\text{opt}} \\ &= \frac{\sum_{j=1}^{\text{opt}} \Delta_{y_j} f(A_i \cup C_{j-1})}{\text{opt}}.\end{aligned}$$

Because f is submodular and monotone increasing, we have

$$\Delta_{y_j} f(A_i) \geq \Delta_{y_j} f(A_i \cup C_{j-1}).$$

Therefore,

$$f(A_{i+1}) - f(A_i) \geq \frac{f(C) - f(A_i)}{\text{opt}}, \tag{2.1}$$

that is,

$$\begin{aligned}f(C) - f(A_{i+1}) &\leq (f(C) - f(A_i)) \left(1 - \frac{1}{\text{opt}}\right) \\ &\leq (f(C) - f(\emptyset)) \left(1 - \frac{1}{\text{opt}}\right)^{i+1} \\ &\leq (f(C) - f(\emptyset)) \mathrm{e}^{-(\mathrm{i}+1)/\text{opt}}.\end{aligned}$$

Choose i such that $f(C) - f(A_{i+1}) < \text{opt} \leq f(C) - f(A_i)$. Then

$$0 = f(C) - f(A_g) < f(C) - f(A_{g-1}) < \cdots < f(C) - f(A_{i+1}) \leq \text{opt} - 1.$$

Therefore,

$$g \leq i + \text{opt}$$

and

$$\text{opt} \leq (f(C) - f(\emptyset)) \mathrm{e}^{-\mathrm{i}/\text{opt}}.$$

Therefore,

$$g \leq \text{opt}\left(1 + \ln \frac{f(C) - f(\emptyset)}{\text{opt}}\right) \leq \text{opt}(1 + \ln \gamma)$$

since

$$f(C) - f(\emptyset) \leq \sum_{j=1}^{\text{opt}} \Delta_{y_j} f(\emptyset) \leq \text{opt} \cdot \gamma. \qquad \square$$

The second observation is that in this analysis, there is only one place that submodularity is required, which is in the proof of inequality (2.1), where we need to have

$$\Delta_{y_j} f(A_i) \geq \Delta_{y_j} f(A_i \cup C_{j-1}).$$

An important observation on this inequality is that the increment variable y_j belongs to optimal solution. Therefore, although for nonsubmodular f this inequality may not holds, we may choose a proper ordering for things in optimal solution to make this inequality almost holds. In the following, we will implement this idea for CDS.

Let vertices $x_1, \ldots, x_g$ be selected in turn by Wolsey Greedy Algorithm. Denote $C_i = \{x_1, x_2, \ldots, x_i\}$ and $a_i = f(C_i)$. Initially, $a_0 = n$ where n is the number of vertices in G. Let C^* be a minimum CDS for G.

Lemma 2.5.2. *For* $i = 1, 2, \ldots, g$,

$$a_i \leq a_{i-1} - \frac{a_{i-1} - 2}{|C^*|} + 1.$$

Proof. First, consider $i \geq 2$. Note that

$$a_i = f(C_i) = a_{i-1} + \Delta_{x_i} f(C_{i-1}),$$

where

$$-\Delta_{x_i} f(C_{i-1}) = \max_y (-\Delta_y f(C_{i-1})).$$

Since C^* is a CDS, we can always arrange elements of C^* in an ordering $y_1, y_2, \ldots, y_{|C^*|}$ such that y_1 is adjacent to a vertex in C_{i-1} and for $j \geq 2$, y_j is adjacent to a vertex in $\{y_1, \ldots, y_{j-1}\}$. Denote $C_j^* = \{y_1, y_2, \ldots, y_j\}$. Then

$$\Delta_{C^*} f(C_{i-1}) = \sum_{j=1}^{|C^*|} \Delta_{y_j} f(C_{i-1} \cup C_{j-1}^*).$$

Note that

$$-\Delta_{y_j} p(C_{i-1} \cup C_{j-1}^*) \leq -\Delta_{y_j} p(C_{i-1}) + 1.$$

In fact, y_j can dominate at most one additional connected component in the subgraph $G[C_{i-1} \cup C_{j-1}^*]$ than in $G[C_{i-1}]$, which is the one contains C_{j-1}^* since $y_1, \ldots, y_{j-1}$ are connected. Moreover, by Lemma 2.4.4,

$$-\Delta_{y_j} q(C_{i-1} \cup C_{j-1}^*) \leq -\Delta_{y_j} q(C_{i-1}).$$

Therefore,

$$-\Delta_{y_j} f(C_{i-1} \cup C_{j-1}^*) \leq -\Delta_{y_j} f(C_{i-1}) + 1.$$

It follows that

$$a_{i-1} - 2 = -\Delta_{C^*} f(C_{i-1}) \\ \leq \sum_{j=1}^{|C^*|} (-\Delta_{y_j} f(C_{i-1}) + 1).$$

There exists $y_j \in C^*$ such that

$$-\Delta_{y_j} f(C_{i-1}) + 1 \geq \frac{a_{i-1} - 2}{|C^*|}.$$

Hence,

$$-\Delta_{x_i} f(C_{i-1}) \geq \frac{a_{i-1} - 2}{|C^*|} - 1.$$

It implies that

$$a_i \leq a_{i-1} - \frac{a_{i-1} - 2}{|C^*|} + 1.$$

For $i = 1$, the proof is similar, we only need to note a difference that y_1 can be chosen arbitrarily. □

Theorem 2.5.3 (Ruan et al. [92]). *Wolsey Greedy Algorithm with $-f = -p - q$ as a potential function gives a polynomial-time $(2 + \ln \delta)$-approximation for* MIN-CDS *where δ is the maximum vertex-degree in input graph.*

Proof. By Lemma 2.5.2,

$$\begin{aligned} a_i - 2 &\leq (a_{i-1} - 2)\left(1 - \frac{1}{|C^*|}\right) + 1 \\ &\leq (a_0 - 2)\left(1 - \frac{1}{|C^*|}\right)^i + \sum_{k=0}^{i-1}\left(1 - \frac{1}{|C^*|}\right)^k \\ &= (a_0 - 2)\left(1 - \frac{1}{|C^*|}\right)^i + |C^*|\left(1 - \left(1 - \frac{1}{|C^*|}\right)^i\right) \\ &= (a_0 - 2 - |C^*|)\left(1 - \frac{1}{|C^*|}\right)^i + |C^*|. \end{aligned}$$

Since $a_i \leq a_{i-1} - 1$ and $a_g = 2$, we have $a_{|C| - 2|C^*|} \geq 2|C^*| + 2$. If $g \leq 2|C^*|$, then we already done the proof. If $g > 2|C^*|$, then set $i = g - 2|C^*|$. Then

$$2|C^*| \leq (n - 2 - |C^*|)\left(1 - \frac{1}{|C^*|}\right)^i + |C^*|.$$

Since $(1-1/|C^*|)^i \le e^{-i/|C^*|}$, we obtain

$$i \le |C^*| \ln \frac{n-2-|C^*|}{|C^*|}.$$

Note that each vertex can dominate at most $\delta+1$ vertices. Hence, $n/|C^*| \le \delta+1$. Therefore, $g = i+2|C^*| \le |C^*|(2+\ln\delta)$. □

Now, let us consider Wolsey Greedy Algorithm for MIN-SET-COVER. If in each iteration we allow to choose two subsets instead of only one subset, could this greedy algorithm get a better performance ratio? The answer is not. This happens also to Wolsey Greedy Algorithm for submodular potential function. However, for nonsubmodular potential function, the situation is changed. The following greedy algorithm will approach to performance ratio $1+\ln\delta$ as $k \to \infty$.

Greedy Algorithm DGPWWZ
input a connected graph G.

Initially, set $C \leftarrow \emptyset$;
while $f(C) > 2$ **do**
choose a subset X of at most $2k-1$ vertices to maximize $-\frac{\Delta_X f(C)}{|X|}$
and set $C \leftarrow C \cup X$;
output $C_g = C$.

To analyze Greedy Algorithm DGPWWZ, we need to note the following property of the potential function $-f$.

Lemma 2.5.4. *Let A and B be two vertex subsets. If both $G[B]$ and $G[X]$ are connected, then*

$$-\Delta_X f(A \cup B) + \Delta_X f(A) \le 1.$$

Proof. Since q is submodular, we have $\Delta_X q(A) \le \Delta_X q(A \cup B)$.

Moreover, since both subgraphs $G[B]$ and $G[X]$ are connected, the number of black components dominated by X in $G[A \cup B]$ is at most one more than the number of black components dominated by X in $G[A]$. Therefore, $-\Delta_X p(A \cup B) \le -\Delta_X p(A) + 1$. Hence, $-\Delta_X f(A \cup B) \le -\Delta_X f(A) + 1$. □

Let C^* be a minimum CDS. We show two properties of C^* in the following two lemmas.

Lemma 2.5.5. *For any integer $k \ge 2$, C^* can be decomposed into $Y_1, Y_2, \ldots, Y_h$ for some natural number h such that*

(a) $C^* = Y_1 \cup Y_2 \cup \cdots \cup Y_h$.
(b) *For $1 \le i \le h$, both $G[Y_1 \cup Y_2 \cup \cdots \cup Y_i]$ and $G[Y_i]$ are connected.*
(c) *$k+1 \le |Y_i| \le 2k-1$ for $1 \le i \le h$ except one, in such an exceptional i, $1 \le |Y_i| \le 2k-1$.*
(d) $|Y_1| + |Y_2| + \cdots + |Y_h| \le |C^*| + h - 1$.

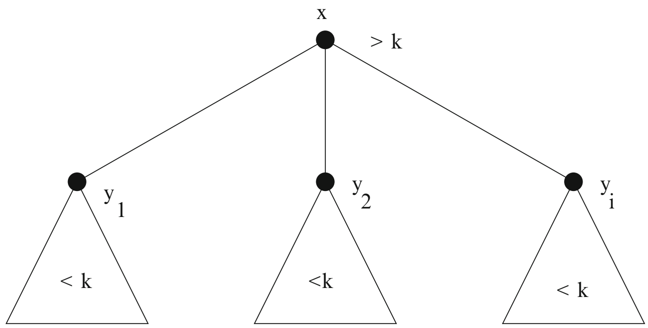


Fig. 2.4 Case 2 in proof of Lemma 2.5.5

Proof. Consider a tree T with vertex set C^*. Choose a vertex $r \in C^*$ as the root of T. For any vertex $x \in C^*$, let $T(x)$ denote the subtree rooted at x and $|T(x)|$ the number of vertices in $T(x)$. If T contains more than $2k-1$ vertices, then there must exist a vertex $x \in C^*$ such that $|T(x)| \geq k+1$ and for every child y of x, $|T(y)| \leq k$. Next, consider two cases.

Case 1. There is a child y of x such that $|T(y)| = k$. Let Y_1 consist of all vertices of $T(y)$ together with x and delete all vertices of $T(y)$ from T.

Case 2. For every child y of x, $|T(y)| \leq k-1$. Suppose $y_1, \ldots, y_i$ are all children of x (Fig. 2.4). There must exist $2 \leq j \leq i$ such that $|T(y_1)| + \cdots + |T(y_j)| \leq k-1$ and $|T(y_1)| + \cdots + |T(y_j)| + |T(y_{j+1})| \geq k$. Since $|T(y_{j+1})| \leq k-1$, we have $|T(y_1)| + \cdots + |T(y_j)| + |T(y_{j+1})| \leq 2k-2$. Let Y_1 consist all vertices in $T(y_1) \cup \cdots T(y_{j+1})$ together with x and delete $Y_1 - \{x\}$ from T.

Repeating above process on the remainder of T, we will obtain a required decomposition. □

Lemma 2.5.6. *Let $n = |V|$. Then, $n \leq (\delta-1)|C^*| + 2$.*

Proof. We prove by induction on $|C^*|$ that C^* with connected $G[C^*]$ can dominate at most $(\delta-1)|C^*| + 2$ vertices. For $|C^*| = 1$, it is trivially true. For $|C^*| \geq 2$, choose a vertex $x \in C^*$ such that $G[C^* - \{x\}]$ is still connected. Removal x would remove at most $\delta - 1$ vertices from the set of vertices dominated by C^*. By the induction hypothesis, $C^* - \{x\}$ can dominate at most $(\delta-1)(|C^*|-1)+2$ vertices. Therefore, C^* can dominate at most $(\delta-1)|C^*| + 2$ vertices. □

Theorem 2.5.7 (Du et al. [40]). *For any $\varepsilon > 0$, there exists a polynomial-time approximation with performance ratio $(1+\varepsilon)\ln(\delta-1)$ for* MIN-CDS.

Proof. Note that the input graph G is connected. If its maximum degree $\delta = 1$, then G contains only one edge. This means that C_g contains only one vertex and is optimal. Hence, Theorem 2.5.7 holds. For $\delta = 2$, G is a path or a cycle. When G is a path, its minimum CDS consists of all internal vertices. When G is a cycle, a minimum CDS can be obtained by deleting two adjacent vertices.

Hence, Theorem 2.5.7 holds. Therefore, we may assume $\delta \geq 3$. Under this assumption, we may further assume $|C_g| > 2|C^*|$ where C^* is a minimum CDS, since, otherwise, $|C_g| \leq 2|C^*| \leq (1+\ln 3)|C^*| \leq (1+c+\ln\delta)|C^*|$.

Suppose $X_1,\ldots,X_g$ are chosen by Greedy Algorithm 2.5 and denote $C_i = X_1 \cup \cdots \cup X_i$. Decompose a minimum CDS C_j^* into $Y_1,\ldots,Y_h$ satisfying conditions in Lemma 2.5.5. Denote $C_j^* = Y_1 \cup \cdots \cup Y_j$. By Lemmas 2.5.4 and 2.5.5,

$$\begin{aligned} -\Delta_{Y_j} f(C_i \cup C_{j-1}^*) &= -\Delta_{Y_j} p(C_i \cup C_{j-1}^*) - \Delta_{Y_j} q(C_i \cup C_{j-1}^*) \\ &\leq -\Delta_{Y_j} p(C_i) + 1 - \Delta_{Y_j} q(C_i) \\ &\leq -\Delta_{Y_j} f(C_i) + 1. \end{aligned}$$

By greedy rule,

$$\frac{-\Delta_{X_{i+1}} f(C_i)}{|X_{i+1}|} \geq \frac{-\Delta_{Y_j} f(C_i)}{|Y_j|} \text{ for } 1 \leq j \leq h.$$

Hence,

$$\begin{aligned} \frac{-\Delta_{X_{i+1}} f(C_i)}{|X_{i+1}|} &\geq \frac{-\sum_{j=1}^h \Delta_{Y_j} f(C_i)}{\sum_{j=1}^h |Y_j|} \\ &\geq \frac{-(h-1) - \sum_{j=1}^h \Delta_{Y_j} f(C_i \cup C_{j-1}^*)}{\sum_{j=1}^h |Y_j|} \\ &\geq \frac{-(h-1) - (f(C_i \cup C^*) - f(C_i))}{\text{opt} + h - 1} \\ &= \frac{f(C_i) - (h+1)}{\text{opt} + h - 1} \end{aligned}$$

where $\text{opt} = |C^*|$. Denote $a_i = f(C_i - (h+1)$. Then,

$$\frac{a_i - a_{i+1}}{|X_{i+1}|} \geq \frac{a_i}{\text{opt} + h - 1},$$

that is,

$$\begin{aligned} a_{i+1} &\leq a_i \left(1 - \frac{|X_{i+1}|}{\text{opt} + h_1}\right) \leq a_i \mathrm{e}^{-|X_{i+1}|/(\text{opt}+h-1)} \\ &\leq a_0 \mathrm{e}^{-(|X_{i+1}|+|X_i|+\cdots+|X_1|)/(\text{opt}+h-1)}. \end{aligned}$$

Choose i such that

$$a_{i+1} < \text{opt} \leq a_i.$$

Denote $b = a_i - \text{opt}$ and $b' = \text{opt} - a_{i+1}$. Write $|X_{i+1}| = d + d'$ such that

$$\frac{b}{d} = \frac{b'}{d'} = \frac{a_i - a_{i+1}}{|X_{i+1}|} \geq \frac{a_i}{\text{opt} + h - 1}.$$

Then we have

$$\frac{a_i - \text{opt}}{d} = \frac{b}{d} \geq \frac{a_i}{\text{opt} + h - 1}.$$

So,

$$\text{opt} \leq a_i \left(1 - \frac{d}{\text{opt} + h - 1}\right) \leq a_i \mathrm{e}^{-d/(\text{opt}+h-1)}.$$

Hence

$$\text{opt} \leq a_0 \mathrm{e}^{-(d+|X_i|+\cdots+|X_1|)/(\text{opt}+h-1)},$$

Note $a_0 = f(\emptyset) - (h+1) = n - (h+1)$. Thus,

$$|X_1| + \cdots + |X_i| + d \leq (\text{opt} + h - 1) \ln \frac{n - (h+1)}{\text{opt}}.$$

Moreover,

$$\begin{aligned} d' + |X_{i+2}| + \cdots + |X_g| &\leq b' + f(C_{i+1}) - f(C_g) \\ &= \text{opt} - a_{i+1} + f(C_{i+1}) - f(C^*) \\ &= \text{opt} + (h-1). \end{aligned}$$

Therefore,

$$|X_1| + \cdots + |X_g| \leq \text{opt}\left(1 + \frac{1}{k}\left(1 + \ln \frac{n - (h+1)}{\text{opt}}\right)\right).$$

By Lemma 2.5.6, $n \leq (\delta - 1)\text{opt} + 2$. Since $h \geq 1$, we have

$$\frac{n - (h+1)}{\text{opt}} \leq \delta - 1.$$

Hence,

$$|X_1| + \cdots + |X_g| \leq \left(1 + \frac{1}{k}\right)(1 + \ln(\delta - 1)).$$

Choose k such that $1/k < \varepsilon$. We obtain Theorem 2.5.7. □

2.6 Weighted CDS

Consider a graph $G = (V,E)$ with weight $w : V \to R^+$. In Chap. 1, we discussed the relationship between CDS and leaves of a spanning tree. From the discussion, we can easily see that a vertex subset is a CDS if and only if it contains all internal vertices of a spanning tree. Therefore, we can obtain the following facts:

- G has a CDS with minimum weight w^* if and only if G has a spanning tree with minimum total internal vertex weight w^*.
- G has a CDS with weight at most w if and only if G has a spanning tree with total internal vertex weight at most w.

Given a digraph and a source vertex s, a *broadcasting tree* is a tree with root s and containing paths that from s to each vertex in the digraph. The broadcasting tree is also called an *out-arborescence*. When we treat G as a digraph by replacing each edge with two arcs with different directions, the following relationship between broadcasting tree and CDS follows from above relationship between spanning tree and CDS:

- G has a CDS with minimum weight w^* if and only if there exists a source vertex s such that G has a broadcasting tree from s with minimum total internal vertex weight w^*.
- G has a CDS with weight at most w if and only if there exists a source vertex s such that G has a broadcasting tree with total internal vertex weight at most w.

Due to this relationship, we first study the following problem on broadcasting tree.

MINW-BROADCAST: Given a digraph $G = (V,E)$ with weight $w : V \to R^+$ and a source node, find a broadcasting tree with minimum total weight of internal nodes.

Consider a subgraph H of input graph G with a source s. An *orphan* of H is a strongly connected component without coming edge and not containing s. Given each node an individual integer ID, the node with smallest ID in an orphan is called the *head* of the orphan.

A *spider* is a subgraph consisting of a body node and several directed paths from the body node to its feet (see Fig. 2.5). A spider is *legal* if it satisfies three conditions:

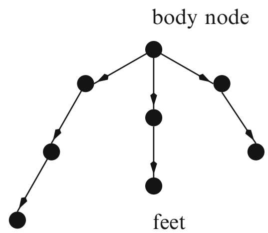


Fig. 2.5 Spider

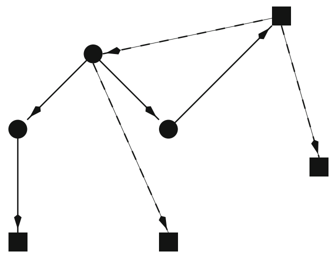


Fig. 2.6 A new orphan is produced by adding a spider

1. All feet are heads of some orphans.
2. S head in it must be a foot or body node.
3. Either its body node is the source s or it contains at least two orphan heads.

We ask for the second condition because putting a legal spider in H may introduce a new orphan at the body node when the body node is not source s so that the number of reduced orphan heads should be the number of orphan heads in it minus one (Fig. 2.6).

For a legal spider S, let $h(S)$ be the number of orphan heads in S and $\text{cost}(S)$ the total weight of internal nodes in S other than internal nodes in H. Define

$$\text{quotient}(S) = \frac{\text{cost}(S)}{h(S)}.$$

For any node u, cut at all orphan heads and consider the connected component C containing u. Suppose $p_1, \dots, p_k$ are k shortest paths from u to k different orphan heads in C. We consider $S = p_1 \cup \dots \cup p_k$ as a spider although $p_1, \dots, p_k$ may have some common nodes other than u. When calculate $\text{cost}(S)$, we assume that all $p_1, \dots, p_k$ are disjoint except at body node. Therefore, $\text{cost}(S)$ is actually an upper bound for the total weight of increased internal nodes. The purpose to make this assumption is to have an easy way to compute $\text{quotient}(v)$ for every node u, which is defined to be

$$\text{quotient}(u) = \min\{\text{quotient}(S) \mid S \text{ is over all legal spider with body node } u\}.$$

With above assumption, $\text{quotient}(u)$ for any node u can be computed in the following way: Suppose H has k orphan heads and $p_1, \dots, p_k$ are shortest paths from node u to them, respectively, ordering that $\text{cost}(p_1) \le \text{cost}(p_2) \le \dots \le \text{cost}(p_k)$. Then for $u \ne s$,

$$\text{quotient}(u) = \min_{2 \le i \le k} \text{quotient}(p_1 \cup \dots \cup p_i),$$

and for $u = s$,

$$\text{quotient}(u) = \min_{1 \le i \le k} \text{quotient}(p_1 \cup \dots \cup p_i).$$

Before state the algorithm, let us show a useful lemma about $\text{quotient}(u)$.

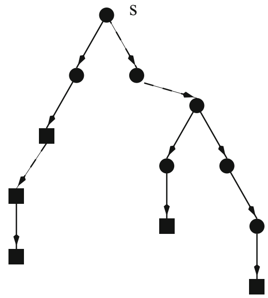


Fig. 2.7 Spider decomposition

Lemma 2.6.1. *Let q be the number of orphans in H. Then there exists a node u with*

$$\text{quotient}(u) \le \frac{\text{opt}}{q},$$

where opt *is the objective function value of optimal solution.*

Proof. Let T^* be an optimal broadcasting tree. We can prune T^* to obtain a subtree T such that every leaf is an orphan head. Now, we can obtain a sequence of legal spiders, $S_1,\dots,S_\ell$ from decomposition of T (Fig. 2.7). Those legal spiders contains all orphan heads and all internal nodes either in H or in T. Therefore,

$$\text{cost}(S_1)+\cdots+\text{cost}(S_\ell) \le \text{opt}$$

and

$$h(S_1)+\cdots+h(S_\ell) = q.$$

Thus,

$$\min_{1\le i\le \ell} \text{quotient}(S_i) \le \frac{\text{opt}}{q}.$$

This means that one of heads for $S_1,\dots,S_\ell$ meet our requirement. □

Algorithm Broadcast:
input a strongly connected digraph $G=(V,E)$ with source node s;
$U \leftarrow \{s\}$;
$O \leftarrow V-\{s\}$;
while $O \ne \emptyset$ **do begin**
    choose node u with smallest quotient cost;
    let $S(u)$ be the legal spider at u reaching quotient(u);
    $U \leftarrow U \cup S(u)$;
    remove from O those orphans whose heads in $S(u)$
    and add back possibly one new orphan;
    recalculate quotient cost of each node;
end-while
output U.

Theorem 2.6.2 (Li et al. [76]). MINW-BROADCAST *has a polynomial-time* $(1+2\ln(n-1))$*-approximation.*

Proof. We analyze the Algorithm Broadcast. Suppose the algorithm runs in k iterations. Initially, there are $n_0 = n-1$ orphans. Let n_i denote the number of orphans right after the ith iteration. For $1 \le i \le k$, let S_i be the legal spider chosen at the ith iteration. Let h_i be the number of heads in S_i and $c_i = \text{cost}(S_i)$. Then

$$n_i \le n_{i-1} - \frac{h_i}{2},$$

since if $h_i = 1$, then

$$n_i \le n_{i-1} - 1 \le n_{i-1} - \frac{h_i}{2};$$

and if $h_i \ge 2$, then

$$n_i \le n_{i-1} - h_i + 1 \le n_{i-1} - \frac{h_i}{2}.$$

Moreover, by Lemma 2.6.1,

$$\frac{c_i}{h_i} \le \frac{\text{opt}}{n_{i-1}}.$$

Thus,

$$\frac{n_i}{n_{i-1}} \le 1 - \frac{c_i}{2\text{opt}}.$$

It implies that

$$\frac{n_{k-1}}{n_0} \le \prod_{i=1}^{k-1} \left(1 - \frac{c_i}{2\text{opt}}\right).$$

Hence,

$$\ln \frac{n_{k-1}}{n_0} \le -\frac{c_1 + \cdots c_{k-1}}{2\text{opt}},$$

that is,

$$c_1 + \cdots + c_{k-1} \le 2\text{opt} \cdot \ln \frac{n_0}{n_{k-1}} \le 2\text{opt} \cdot \ln(n-1).$$

Since $\frac{c_k}{h_k} \le \frac{\text{opt}}{n_{k-1}}$ and $h_k = n_{k-1}$, we have $c_k \le \text{opt}$. Therefore,

$$c_1 + \cdots + c_k \le (1 + 2\ln(n-1)) \cdot \text{opt}.$$

□

Now, we return to MINW-CDS.

Theorem 2.6.3. MINW-CDS *has a polynomial-time* $(1+2\ln(n-1))$*-approximation where n is the number of nodes in input graph.*

Proof. Suppose $G = (V,E)$ is an input graph with weight $w : V \to R^+$. Choose a node $u \in V$. Let $N(u)$ denote the set of neighbors of u and u. For each $v \in N(u)$, compute a broadcasting tree T_v with source v by Algorithm Broadcast. From those T_v for $v \in N(u)$, choose T_{v^*} with minimum total weight of internal nodes. We show that all internal nodes of T_{v^*} form a CDS C with total weight within a factor of $1+2\ln(n-1)$ from optimal.

Let C^* be a CDS with minimum total weight w^*. Note that $C^* \cap N(u) \neq \emptyset$. Choose $v \in C^* \cap N(u)$. Construct a spanning tree for $G[C^*]$ and extend it to a spanning tree for G. Give each edge a direction to form a broadcasting tree T_v^* from source v. Then, T_v^* has total internal node weight at most w^*. By Theorem 2.6.2,

$$\text{weight}(T_v^*) \leq (1+2\ln(n-1))\text{weight}(T_v^*),$$

where weight(T_v) denotes the total internal node weight of T_v. Therefore,

$$\text{weight}(C) \leq \text{weight}(T_v^*) \leq (1+2\ln(n-1))\text{weight}(T_v^*) \leq (1+2\ln(n-1))w^*. \qquad \square$$

2.7 Directed CDS

In this section, we show a relationship between SCDS and the broadcast tree.

Lemma 2.7.1. *Let* $\text{opt}_{\text{BT}}(G,r)$ *be the objective function value of optimal solution for* MINW-BROADCAST *on input G and a source r. Let* $\text{opt}_{\text{SCDS}}(G)$ *be the objective function value of an optimal solution for* MIN-SCDS *on input G. Then for any r,*

$$\text{opt}_{\text{BT}}(G,r) \leq \text{opt}_{\text{SCDS}}(G).$$

Moreover, if r belongs to an optimal solution for MIN-SCDS*, then*

$$\text{opt}_{\text{BT}}(G,r) \leq \text{opt}_{\text{SCDS}}(G) - 1.$$

Proof. Let C^* be the minimum SCDS of G. For any resource r, we can first get in C^* and then through C^* to reach other nodes not in C^* so that the broadcasting tree uses only nodes in C^* as internal nodes except r. When $r \in C^*$, r can be taken off in counting $\text{opt}_{\text{BT}}(G,r)$. $\square$

Lemma 2.7.2. *If there exists polynomial-time α-approximation for* MINW-BROADCAST*, then there exists polynomial-time 2α-approximation for* MIN-SCDS.

Proof. Let G^R be a directed graph obtained from G by reversing the direction of each edge. Let C^* be a minimum strongly SCDS. Choose a node u arbitrarily. Let $N(u)$ be the set consisting of the node u and its in-neighbors, that is those nodes each of which has an edge coming to u. Clearly, $N(u) \cap C^* \neq \emptyset$. For each $s \in N(u)$, compute an α-approximation T_1 for MINW-BROADCAST on input G and source s and also a α-approximation T_2^R for MINW-BROADCAST on input G^R with source s. Let T_2 be the tree obtained from T_2^R by reversing the direction of each edge. Then $T_1 \cup T_2$ is a strongly connected spanning subgraph of G. Furthermore, $I(T_1) \cup I(T_2^R)$ induced a strongly connected subgraph of $T_1 \cup T_2$, dominating G. Hence $I(T_1) \cup I(T_2^R)$ is a SCDS for G where $I(T_i)$ denotes the set of internal nodes in T_i. Clearly,

$$\begin{aligned}|I(T_1) \cup I(T_2)| &\leq |I(T_1) - \{s\}| + |I(T_2) - \{s\}| + |\{s\}| \\ &\leq \alpha(\text{opt}_{\text{BT}}(G,s) + \text{opt}_{\text{BT}}(G^R,s)) + 1.\end{aligned}$$

Note that when s belongs to a minimum SCDS, we would have

$$\begin{aligned}&\alpha(\text{opt}_{\text{BT}}(G,s) + \text{opt}_{\text{BT}}(G^R,s) + 1 \\ &\quad\leq \alpha(\text{opt}_{\text{SCDS}}(G) - 1 + \text{opt}_{\text{SCDS}}(G^R) - 1) + 1 \\ &\quad\leq 2\alpha \cdot \text{opt}_{\text{SCDS}}(G),\end{aligned}$$

since a minimum SCDS for G is also a minimum SCDS for G^R, vice versa. Now, for s over all nodes in $N(u)$, we choose the one such that $I(T_1) \cup I(T_2^R)$ has the smallest cardinality. Such a $I(T_1) \cup I(T_2^R)$ will have cardinality upper bounded by $2\alpha \cdot \text{opt}_{\text{SCDS}}(G)$. □

By Lemma 2.7.2 and Theorem 2.6.2, we have

Theorem 2.7.3 (Li et al. [76]). *There exists a polynomial-time* $(2 + 4\ln(n-1))$-*approximation for* MIN-SCDS.