Analysis of Greedy Approximations with Nonsubmodular Potential Functions

Ding-Zhu Du* Ronald L. Graham† Panos M. Pardalos ‡ Peng-Jun Wan §
Weili Wu ¶ Wenbo Zhao ‖

Abstract

In this paper, we present two techniques to analyze greedy approximation with nonsubmodular functions restricted submodularity and shifted submodularity. As an application of the restricted submodularity, we present a worst-case analysis of a greedy algorithm for Network Steiner tree adapted from a heuristic originally proposed by Chang in 1972 for Euclidean Steiner tree. The performance ratio of Chang's heuristic is a long-standing open problem due to the nonsubmodularity of its potential function. As an application of the shifted submodularity, we present a worst-case analysis of a greedy algorithm for Connected Dominating Set generalized from a greedy algorithm proposed by Ruan *et al.* Such generalized greedy algorithm is shown to have performance ratio at most $(1+\varepsilon)(1+\ln(\Delta-1))$, which matches the well-known lower bound $(1-\varepsilon)\ln\Delta$, where Δ is the maximum vertex-degree of input graph and ε is any positive constant.

1 Introduction

Greedy strategy is one of the major techniques in the design of approximation algorithms for NP-hard optimization problems. If the potential function used by a greedy approximation algorithm is submodular, the performance ratio of the greedy approximation algorithm can be easily derived. For greedy approximations with nonsubmodular potential functions, their performance analysis is a largely unexplored open area. Indeed, many greedy heuristics with good performance demonstrated in computational experiments cannot receive a theoretical analysis due to the difficulty on dealing with nonsubmodular potential function.

In this paper, we introduce two techniques to analyze greedy approximations with nonsubmodular potential functions: restricted submodularity and shifted submodularity. These two techniques are illustrated by their applications in analyzing two greedy approximation algorithms for Network Steiner Tree and Connected Dominating Set respectively. The greedy algorithm for Network Steiner tree to be studied in this paper is adapted from a heuristic originally proposed by Chang [1, 2] in 1972 for Euclidean Steiner tree. Chang's heuristic starts with a minimum spanning tree and at each iteration chooses a Steiner point such that using this Steiner point to connect three vertices in the current tree could replace two edges in the minimum spanning tree and this replacement achieves the maximum gain among such possible replacements. The performance ratio of Chang's heuristic is a long-standing open problem due to the nonsubmodularity of its potential function. A "submodular" variant of Chang's heuristic was proposed by Zelikovsky [8] for Network Steiner Tree. Zelikovsky's heuristic allows the newly added Steiner vertex to connect only three terminals and adopts a submodular gain function to choose the Steiner vertex in each iteration. As a result, Zelikovsky [8] was able to establish a 11/6 bound on the performance ratio of his heuristic. In this paper, we prove, by using the technique of *restricted submodularity*, that Chang's heuristic adapted for Network Steiner Tree also has performance ratio bounded by 11/6.

The greedy algorithm for Connected Dominating Set to be studied in this paper is a generalization of the greedy algorithm proposed by Ruan *et al.* [6] which uses a nonsubmodular potential function. The latter one chooses at each iteration a single vertex with maxi-

*Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA and Xi'an Jiaotong University, Xi'an, China. Email: dzdu@utdallas.edu. Support in part by the NSF under grant CCF-9208913.

†Department of Computer Science, University of California at San Diego, La Jolla, CA 92093, USA. Email: graham@ucsd.edu.

‡Department of Industrial Engineering and System Science, University of Florida, Gainsville, FL 32611, USA. Email: pardalos@ufl.edu.

§Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616, USA. Email: wan@cs.iit.edu.

¶Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA. Email weiliwu@utdallas.edu. Support in part by the NSF under grant CCF-9208913.

‖Department of Computer Science and Engineering, University of California at San Diego La Jolla, CA 92093, USA. Email: w3zhao@cs.ucsd.edu.

mum gain, and has a performance ratio at most $2+\ln\Delta$, where Δ is the maximum vertex-degree of the input graph. Our generalization chooses at each iteration a set of up to $2k-1$ vertices with the maximum gain-cost ratio, where k is a positive integer parameter. Using the technique of *shifted submodularity*, we prove that such generalization has a performance ratio at most $\left(1+\frac{1}{k}\right)(1+\ln(\Delta-1))$. This implies that for any $\varepsilon>0$ there is a $(1+\varepsilon)(1+\ln(\Delta-1))$-approximation algorithm for Connected Dominating Set. An interesting observation is that for greedy approximation algorithms with submodular potential functions, the above generalization cannot lead to better performance ratio.

2 Minimum Submodular Cover

Consider a ground set E and a real function f defined on 2^E. f is *increasing* if $A\subset B$ implies $f(A)\leq f(B)$. f is *submodular* if for any two subsets A and B of E,

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

The *marginal value* of $B\subseteq E$ with respect to $A\subseteq E$ is defined by

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

Similarly, the *marginal value* of an element $e\in E$ with respect to $A\subseteq E$ is defined by

$$\Delta_e f(A)=f(A\cup\{e\})-f(A).$$

Both monotonicity and submodularity of a function f can be characterized in terms of the marginal values (see, e.g., [5]): f is *increasing* if $\Delta_x f(A)\geq 0$ for any $A\subseteq E$ and $x\in E\setminus A$; f is submodular if and only if for any $A\subseteq E$ and *different* $x,y\in E\setminus A$, the inequality $\Delta_x f(A)\geq\Delta_x f(A\cup\{y\})$ holds. In addition, the following are equivalent:

- f is increasing and submodular.
- For any $A,B\subseteq E$,
$$f(B)-f(A)\leq\sum_{x\in B\setminus A}\Delta_x f(A).$$
- For any $A\subseteq E$ and $x,y\in E\setminus A$,
$$\Delta_x f(A)\geq\Delta_x f(A\cup\{y\}).$$

Suppose that c is a nonnegative cost function on a ground set E, and f is an integer-valued, increasing and submodular function on 2^E. The minimization problem

$$\min\left\{\sum_{x\in A}c(x):f(A)=f(E),A\subseteq E\right\}$$

is known as **Minimum Submodular Cover**. A greedy approximation for it is as follows.

Greedy Algorithm GSC
$A\leftarrow\emptyset$;
While $\exists e\in E$ such that $\Delta_e f(A)>0$ do
  select $a\in E$ with maximum $\Delta_a f(A)/c(a)$;
  $A\leftarrow A\cup\{a\}$;
Output A.

A general result on greedy algorithms with increasing submodular potential functions has been existing in the literature for a long time .

THEOREM 2.1. *Greedy Algorithm **GSC** produces an $H(\gamma)$-approximation solution for Minimum Submodular Cover, where $\gamma=\max_{x\in E}\Delta_x f(\emptyset)$ and $H(k)$ is the k-th Harmonic number ([7]).*

For many specific **Minimum Submodular Cover** problems, the potential function f is not given explicitly. For these problems, the main task is to find the proper submodular potential function. Some **Minimum Submodular Cover** problems such as Set Cover, the submodular potential functions can be easily defined. But for some others, it is far from trivial to find the submodular potential function. A prominent example is Weighted Connected Vertex Cover (CVC). An instance of Weighted CVC is a connected vertex-weighted graph $G=(V,E)$. A solution to Weighted CVC is a connected vertex cover, which is a subset of vertices inducing a connected subgraph and covering all edges. The objective is to find a connected vertex-cover with the minimum total weight. The weighted CVC has the same approximation hardness as weighted set cover [4]. The best-known approximation algorithm for Weighted CVC is a two-phased $(2+H(\Delta-1))$-algorithm presented in [4], where Δ is the maximum degree of G. In the next, we show that Weighted CVC can be formulated as a **Minimum Submodular Cover** problem, and consequently the greedy approximation algorithm can be applied. We pick V as the ground set. For any subset C of V, define $g(C)$ to be the number of edges in G covered by C, $h(C)$ to be the number of connected components of $G[C]$, and $f(C)=g(C)-h(C)$. Clearly, $f(\emptyset)=0$, and $f(V)=|E|-1$. In addition, f has the following properties.

LEMMA 2.1. *If $|E|>1$ then C is a connected vertex-cover if and only $f(C)=|E|-1$. Furthermore, f is increasing and submodular.*

Proof. If C is a connected vertex-cover, then $f(C)=g(C)-h(C)=|E|-1$. Conversely, suppose that $f(C)=|E|-1$. Since $|E|>1$, C is nonempty and hence $h(C)\geq 1$. On the other hand, $g(C)\leq|E|$. So, we must have $g(C)=|E|$ and $h(C)=1$. The former implies that C is a vertex-cover, while the latter implies that C induces a connected subgraph. Hence C is a connected vertex-cover.

Now, we prove that f is increasing. Consider a vertex subset C and a vertex $u \notin C$. We show that $\Delta_u f(C) \geq 0$. We denote by $N(u)$ the set of neighbors of u in G. Then,

$$\Delta_u g(C) = |N(u) \setminus C| \geq 0.$$

On the other hand, $-\Delta_u h(C)$ is equal to the number of connected components in $G[C]$ adjacent to u minus one. If $C \cap N(u) = \emptyset$, then $\Delta_u g(C) = \deg(u)$ and $-\Delta_u h(C) = -1$, which implies that $\Delta_u f(C) = \deg(u) - 1 \geq 0$. If $C \cap N(u) \neq \emptyset$, then u is adjacent to at least one connected component of $G[C]$ and hence $-\Delta_u h(C) \geq 0$, which also implies that $\Delta_u f(C) \geq 0$.

Next, we prove that f is submodular. Consider a vertex subset C and two different vertices u and v in $V \setminus C$. We show that $\Delta_u f(C \cup \{v\}) \leq \Delta_u f(C)$ in two cases:

Case 1: u is not adjacent to v. Then, $\Delta_u g(C \cup \{v\}) = \Delta_u g(C)$. Consider an arbitrary connected component of $G[C \cup \{v\}]$ adjacent to u. If it does not contain v, then it is also a connected component of $G[C]$ adjacent to u. If it contains v, then it must contain at least one connected component of $G[C]$ adjacent to u. Thus, the number of connected components of $G[C \cup \{v\}]$ adjacent to u is no more than the number of connected components of $G[C]$ adjacent to u. In other words, $-\Delta_u h(C \cup \{v\}) \leq -\Delta_u h(C)$. So $\Delta_u f(C \cup \{v\}) \leq \Delta_u f(C)$.

Case 2: u is adjacent to v. Then, $\Delta_u g(C \cup \{v\}) = \Delta_u g(C) - 1$. Among all connected components of $G[C \cup \{v\}]$ adjacent to u, exactly one contains v and all others are each a connected component of $G[C]$ adjacent to u. Hence, $-\Delta_u h(C \cup \{v\}) \leq -\Delta_u h(C) + 1$. Therefore, $\Delta_u f(C \cup \{v\}) \leq \Delta_u f(C)$.

Note that

$$\max_{u \in V} \Delta_u f(\emptyset) = \max_{u \in V} (\deg(u) - 1) = \Delta - 1.$$

Greedy Algorithm **GSC** with potential function f as defined above is a $H(\Delta - 1)$-approximation for Weighted CVC.

3 Network Steiner Tree

An instance of Network Steiner Tree consists of a complete graph G with metric cost (or length) over a set V of vertices and a subset R of vertices, called *terminals*, in V. The objective is is to compute a shortest tree in G interconnecting all terminals. While a minimum solution is called a *Steiner minimum tree*, every tree interconnecting all terminals without nonterminal leaf is called a *Steiner tree*. A Steiner tree may contain some vertices other than terminals. Those vertices are called *Steiner vertices*. In this section, we first describe a heuristic for Network Steiner Tree adapted from Chang's heuristic for Euclidean Steiner tree. After that, we give a worst-case analysis of this heuristic which illustrates the technique of restricted submodularity.

Denote by $G[R]$ the subgraph of G induced by R. Consider a subgraph H of G in which each connected component contains at least one terminal. The union of minimum spanning trees in all connected components of H is denoted by $MSF(H)$, and its length is denoted by $msf(H)$. A forest F in $G[R]$ is called a spanning tree on $(R : H)$ if $MSF(H) \cup F$ is a tree spanning all terminals in R. We denote by $\overline{H}$ the graph with vertex set $V(H) \cup R$ and edge set $E(H)$. Then, a spanning tree on $(R : H)$ can be regarded as a spanning tree on the connected components of $\overline{H}$. We denote by $MST(R : H)$ a minimum spanning tree on $(R : H)$, and by $mst(R : H)$ the length of $MST(R : H)$. Equivalently, $MST(R : H)$ is a minimum spanning tree on R after every component of H is contracted into a vertex. Define

$$g(H) = mst(R) - mst(R : H),$$
$$f(H) = mst(R) - mst(R : H) - msf(H).$$

A star in G is *legal* w.r.t. H if it is either an edge, or a 2-star, or a 3-star joining at most three connected components of $\overline{H}$. The following is a greedy version of Chang's heuristic:

Greedy Algorithm CST
$H \leftarrow \emptyset$;
While $mst(R : H) > 0$ do
    choose a legal (w.r.t. H) star T
        maximizing $\Delta_T f(H)$;
    $H \leftarrow H \cup T$;
Output $MST(H)$.

THEOREM 3.1. *Greedy Algorithm* ***CST*** *is a 11/6-approximation for Network Steiner Tree.*

We begin with some definitions and notations. A tree in G is said to be *full* if all its leaves are terminals and none of its internal nodes is a terminal. A full tree is said to be k-restricted if it contains at most k terminals. Consider a tree T without non-terminal leaf. We decompose T into full subtrees at terminals with degree more than one such that in each subtree every terminal becomes a leaf. Those subtrees are called *full components* of T. T is said to be k-restricted if all of its full components are k-restricted. A forest without non-terminal leaf is said to be k-restricted if all of its tree components are k-restricted. Suppose that S is the edge-disjoint union of full trees $S_1, S_2, \cdots, S_l$. For each $1 \le i \le l$, let T_i be a spanning tree of the subgraph of G induced by the terminals in S_i. If $T_1 \cup T_2 \cup \cdots \cup T_l$ is a spanning tree of $G[R]$, then S is called a *Steiner hypertree*, and each S_i is called a full component of S. The cost of a Steiner hypertree is defined to be the total costs of its full components. A Steiner hypertree is said to be k-restricted if all of its full components are k-restricted. In general, a Steiner hypertree may not be a Steiner tree, and it is a Steiner tree if and only if any pair of its full components have no common Steiner vertex.

Consider a 3-restricted Steiner minimum hypertree SMT_3 in G with the smallest number of Steiner points. Clearly, each full component of SMT_3 is either an edge or a 3-star. Let T be the Steiner tree output by the algorithm **CST**. It's possible that SMT_3 may use some Steiner vertices in T, and some full components of SMT_3 may have a common Steiner vertex. For each full component of SMT_3 whose Steiner vertex belongs to T, we add to G an exclusive replication of such full component (but without duplicating terminals); for each Steiner vertex which is not in T but is shared by multiple full components, we add to G exclusive replications of all but one of these full components. Denote by G^+ the union of G and the replicated full components. Replacing each full component of SMT_3 whose Steiner vertex belongs to T or is shared by others by its replication, we obtain a 3-restricted Steiner minimum tree SMT_3^+ in G^+, which contains no Steiner vertex in T, and has the same cost as SMT_3.

Suppose that H is a subgraph of G in which each connected component contains at least two terminals. A forest F in G^+ is called a Steiner forest on $(R : H)$ if $MSF(H) \cup F$ is a forest, all its leaves are terminals, and none of its Steiner vertices belongs to H. A Steiner forest F on $(R : H)$ is called a Steiner tree on $(R : H)$ if $MSF(H) \cup F$ is a tree interconnecting all terminals. Then, for any Steiner forest F on $(R : H)$,

$$MSF(H \cup F) = MSF(H) \cup F.$$

We extend the domain of $g(\cdot)$ and $f(\cdot)$ to the set of subgraphs of G^+. Clearly, $g(\cdot)$ is a monotone increasing function on the set of subgraphs of G^+, and $g(H) = mst(R)$ if and only if H is a connected graph interconnecting all terminals. It's also well-known that $g(\cdot)$ is submodular when restricted on the set of subgraphs of $G[R]$ (see, e.g., [8]). The next lemma is a generalization of this restricted submodularity.

Lemma 3.1. *Suppose that H is a subgraph of G and S is Steiner tree on $(R : H)$ in G^+. Let g^* and f^* be the functions on the set of subgraphs of G defined by $g^*(A) = g(H \cup A)$ and $f^*(A) = f(H \cup A)$ respectively. Then both of them are submodular restricted on the set of unions of full components of S.*

Proof. Suppose that $H_1, H_2, \cdots, H_l$ are the connected components of H. For each $1 \le i \le l$, let H'_i be a spanning tree of the subgraph of G induced by the terminals in H_i. Denote

$$H' = H'_1 \cup H'_2 \cup \cdots \cup H'_l.$$

Suppose that $S_1, S_2, \cdots, S_m$ are the full components of S. For each $1 \le i \le m$, let S'_i be a spanning tree of the subgraph of G induced by the terminals in S_i. For any graph $A = \cup_{i \in I} S_i$ for some $I \subseteq \{1, 2, \cdots, m\}$, denote $A' = \cup_{i \in I} S'_i$. Then, for any subgraph A which is the union of some full components of S,

$$g^*(A) = g(H \cup A) = g(H' \cup A').$$

Now suppose that A and B are any two graphs which are unions of some full components of S. Since g is submodular on the set of subgraphs of $G[R]$, we have

$$\begin{aligned}
& g^*(A) + g^*(B) = g(H' \cup A') + g(H' \cup B') \\
& \ge g((H' \cup A') \cup (H' \cup B')) + g((H' \cup A') \cap (H' \cup B')) \\
& = g(H' \cup (A' \cup B')) + g(H' \cup (A' \cap B')) \\
& = g^*(A \cup B) + g^*(A \cap B).
\end{aligned}$$

Therefore, g^* submodular restricted on the set of unions of the full components of S.

Note that for any subgraph A of S,

$$msf(H \cup A) = msf(H) + msf(A).$$

Thus,

$$f^*(A) = g^*(A) - msf(H) - msf(A).$$

Clearly, $msf(\cdot)$ is modular restricted on the set of unions of the full components of S. Hence, f^* is also submodular on the set of unions of the full components of S.

Corollary 3.1. *Suppose that H is a subgraph of G, S is Steiner tree on $(R : H)$ in G^+ with full components $\{S_1, S_2, \cdots, S_m\}$ and l is a positive integer at most m. For each $1 \le i \le l$, let F_i be a forest in the subgraph of G induced by the terminals in S_i. Then,*

$$\begin{aligned}
& \Delta_{S_{l+1} \cup \cdots \cup S_m} g(H \cup F_1 \cup \cdots \cup F_l) \\
& \ge \Delta_{S_{l+1} \cup \cdots \cup S_m} g(H \cup S_1 \cup \cdots \cup S_l).
\end{aligned}$$

Proof. For each $1 \leq i \leq l$, let S'_i be a spanning tree in the subgraph of G induced by the terminals in S_i which contains F_i. Then, the union of H'_i with $1 \leq i \leq l$ and S_i with $l+1 \leq i \leq m$ is also a Steiner tree on $(R : H)$. By Lemma 3.1,

$$\Delta_{S_{l+1}\cup\cdots\cup S_m} f(H \cup F_1 \cup \cdots \cup F_l) \\ \geq \Delta_{S_{l+1}\cup\cdots\cup S_m} f(H \cup S'_1 \cup \cdots \cup S'_l).$$

On the other hand,

$$\Delta_{S_{l+1}\cup\cdots\cup S_m} g(H \cup S'_1 \cup \cdots \cup S'_l) \\ \geq \Delta_{S_{l+1}\cup\cdots\cup S_m} g(H \cup S_1 \cup \cdots \cup S_l).$$

Thus, the corollary follows.

LEMMA 3.2. *Suppose that H is a subgraph of G and S is Steiner tree on $(R : H)$ in G^+. Then, for each full component Q of S, there is a legal (w.r.t. H) star $Q' \subseteq G$ satisfying that $\Delta_Q f(H) \leq \Delta_{Q'} f(H)$.*

Proof. The lemma is trivial if $Q \subseteq G$, and so we assume that $Q \subsetneq G$. Then Q is a replicated from some 3-star $\{ou, ov, ow\}$. If $o \notin V(H)$, set Q' to be this 3-star. Then Q' is a legal star w.r.t. H, and $\Delta_{Q'} f(H) = \Delta_Q f(H)$. So, we further assume $o \in V(H)$. Suppose that

$$MST(R : H \cup Q) = MST(R : H) - \{e^*, e^{**}\}.$$

Then,

$$\Delta_Q g(H) = length(e^*) + length(e^{**}),$$

and $MST(H) \cup MST(R : H) - \{e^*, e^{**}\}$ consists of three tree components containing u, v, and w respectively. Without loss of generality, suppose that o lies in the tree component containing u. Set $Q' = \{ov, ow\}$. Then Q' is a legal, and $MST(H) \cup MST(R : H) - \{e^*, e^{**}\} \cup Q'$ is also a tree interconnecting all the terminals. Hence, $MST(R : H) - \{e^*, e^{**}\} \cup Q'$ is a spanning tree on $(R : H)$, which implies that

$$\Delta_{Q'} g(H) \geq length(e^*) + length(e^{**}) = \Delta_Q g(H).$$

Clearly,

$$msf(H \cup Q') = msf(H) + msf(Q').$$

Therefore,

$$\Delta_{Q'} f(H) = \Delta_{Q'} g(H) - msf(Q') \\ \geq \Delta_Q g(H) - msf(Q) = \Delta_Q f(H).$$

Consider the following greedy algorithm which is a generalization to **CST**:

```
Greedy Algorithm CST(A)
H ← A;
While mst(R : H) > 0 do
    choose a legal (w.r.t. H) star T
        maximizing Δ_T f(H);
    H ← H ∪ T;
Output MST(H).
```

LEMMA 3.3. *Suppose that H is a subgraph of G in which every connected component contains at least two terminals, and T is the Steiner tree obtained by Greedy Algorithm **CST**(H). Then, for any 3-restricted Steiner tree S on $(R : H)$ in G^+ which contains no Steiner point in T, $2\Delta_T f(H) \geq \Delta_S f(H)$.*

Proof. Let $T_1, T_2, \cdots, T_k$ be the sequence of trees chosen by Greedy Algorithm **CST**(H) in the order of appearance. We prove the lemma by induction on k. If $k = 0$, then both T and S are empty and the lemma holds trivially. Now assume that $k > 0$ and denote $T' = T_2 \cup \cdots \cup T_k$. Note that

$$\Delta_S f(H) = mst(P : H) + mst(H) - mst(H \cup S) \\ = mst(P : H) - mst(S).$$

It is sufficient to prove that the lemma holds when S is the shortest 3-restricted Steiner tree on $(R : H)$. So we assume that S is a shortest 3-restricted Steiner tree on $(R : H)$. Then, each full component of S is either an edge or a star with three leaves. By Lemma 3.2 and the greedy principle, $\Delta_Q f(H) \leq \Delta_{T_1} f(H)$ for each full component Q of S. The rest of the proof is divided into three cases depending on the number of connected components of $\overline{H}$ joined by T_1.

Case 1. T_1 joins only one connected components of $\overline{H}$.

In this case, S is still a 3-restricted Steiner tree on $(R : H \cup T_1)$. By induction hypothesis,

$$2\Delta_{T'} f(H \cup T_1) \geq \Delta_S f(H \cup T_1).$$

On the other hand,

$$\Delta_S f(H \cup T_1) = \Delta_S g(H \cup T_1) - mst(S) \\ = \Delta_S g(H) - mst(S) = \Delta_S f(H).$$

Therefore, by the greedy principle,

$$2\Delta_T f(H) = 2\Delta_{T'} f(H \cup T_1) + 2\Delta_{T_1} f(H) \\ \geq \Delta_S f(H \cup T_1) = \Delta_S f(H).$$

Case 2. T_1 joins two connected components of $\overline{H}$.

We first claim that S contains a full component S_1 satisfying that $S' = S - S_1$ is a Steiner forest on $(R : H \cup T_1)$ and

$$\Delta_{S'} f(H \cup T_1) \geq \Delta_{S'} f(H \cup S_1).$$

Indeed, pick two terminals u and v from the two connected components of $\overline{H}$ joined by T_1 respectively. Let $e' = uv$ and suppose that

$$MST(R : H \cup T_1) = MST(R : H) - e^*.$$

Then,

$$\Delta_{T_1} g(H) = length(e^*).$$

Let P_1 be the path in $MST(H) \cup S$ between u and v. Connect every pair of consecutive terminals in each path component of $P_1 \cap S$ with an edge. Then, there exists one such edge e_1 bridging the two tree components of $MST(H) \cup MST(R : H) - e^*$. So, $MST(H) \cup MST(R : H) - e^* + e_1$ is also a tree interconnecting all terminals. Hence,

$$\Delta_{e_1} g(H) \geq length(e^*) = \Delta_{T_1} g(H).$$

Suppose that S_1 is the full component of S containing two endpoints of e_1, and let $S' = S - S_1$. Then $MST(H) \cup S' \cup \{e'\}$ is a forest and so is $MST(H) \cup S' \cup T_1$. In addition, $MST(H) \cup S' \cup \{e_1\}$ and $MST(H) \cup S' \cup \{e'\}$ have the same collection of components, which implies that

$$\Delta_{S' \cup \{e_1\}} g(H) = \Delta_{S' \cup \{e'\}} g(H) = \Delta_{S' \cup T_1} g(H).$$

Thus,

$$\begin{aligned} &\Delta_{S'} g(H \cup \{e_1, e_2\}) = \Delta_{S' \cup \{e_1\}} g(H) - \Delta_{e_1} g(H) \\ &\leq \Delta_{S' \cup T_1} g(H) - \Delta_{T_1} g(H) = \Delta_{S'} g(H \cup T_1). \end{aligned}$$

By Corollary 3.1,

$$\Delta_{S'} g(H \cup \{e_1\}) \geq \Delta_{S'} g(H \cup S_1),$$

Therefore,

$$\Delta_{S'} g(H \cup T_1) \geq \Delta_{S'} g(H \cup S_1 \cup S_2).$$

Hence,

$$\begin{aligned} &\Delta_{S'} f(H \cup T_1) = \Delta_{S'} g(H \cup T_1) - msf(S') \\ &\geq \Delta_{S'} g(H \cup S_1) - msf(S') = \Delta_{S'} f(H \cup S_1). \end{aligned}$$

Thus, our claim holds.

Note that $S' \cup MST(R : H \cup T_1 \cup S')$ is a 3-restricted Steiner tree on $(R : H \cup T_1)$ and contains no Steiner point in T'. By induction hypothesis, we have

$$\begin{aligned} &2\Delta_{T'} f(H \cup T_1) \geq \Delta_{S' \cup MST(R:H \cup T_1 \cup S')} f(H \cup T_1) \\ &= mst(R : H \cup T_1) - msf(S') - mst(R : H \cup T_1 \cup S') \\ &= \Delta_{S'} f(H \cup T_1) \geq \Delta_{S'} f(H \cup S_1). \end{aligned}$$

Therefore, by the greedy principle,

$$\begin{aligned} &2\Delta_T f(H) = 2\Delta_{T'} f(H \cup T_1) + 2\Delta_{T_1} f(H) \\ &\geq \Delta_{S'} f(H \cup S_1) + \Delta_{S_1} f(H) + \Delta_{T_1} f(H) \\ &= \Delta_S f(H) + \Delta_{T_1} f(H) \geq \Delta_S f(H). \end{aligned}$$

Case 3. T_1 joins three connected components of $\overline{H}$.

Similarly, we can prove that that S contains two (possibly identical) full components S_1 and S_2 satisfying that $S' = S - (S_1 \cup S_2)$ is a Steiner forest on $(R : H \cup T_1)$ and

$$\Delta_{S'} f(H \cup T_1) \geq \Delta_{S'} f(H \cup S_1 \cup S_2).$$

So, $S' \cup MST(R : H \cup T_1 \cup S')$ is a 3-restricted Steiner tree on $(R : H \cup T_1)$ and contains no Steiner point in T'. By induction hypothesis, we have

$$\begin{aligned} &2\Delta_{T'} f(H \cup T_1) \geq \Delta_{S' \cup MST(R:H \cup T_1 \cup S')} f(H \cup T_1) \\ &= mst(R : H \cup T_1) - msf(S') - mst(R : H \cup T_1 \cup S') \\ &= \Delta_{S'} f(H \cup T_1) \geq \Delta_{S'} f(H \cup S_1 \cup S_2). \end{aligned}$$

By greedy principle,

$$\begin{aligned} &2\Delta_T f(H) = 2\Delta_{T'} f(H \cup T_1) + 2\Delta_{T_1} f(H) \\ &\geq \Delta_{S'} f(H \cup S_1 \cup S_2) + \Delta_{S_1} f(H) + \Delta_{S_2} f(H) \\ &\geq \Delta_{S'} f(H \cup S_1 \cup S_2) + \Delta_{S_1 \cup S_2} f(H) = \Delta_S f(H). \end{aligned}$$

This completes the proof.

Now we are ready to prove Theorem 3.1. Note that Greedy Algorithm **CST** is exactly **CST**($\emptyset$). Let T be the Steiner tree obtained from Greedy Algorithm **CST**. By Lemma 3.3,

$$2\Delta_T f(\emptyset) \geq \Delta_{SMT_3^+(R)} f(\emptyset).$$

which implies that

$$2f(T) \geq f(SMT_3^+(R)) = f(SMT_3(R)).$$

Hence,

$$length(T) \leq \frac{1}{2}(mst(R) + smt_3(R)).$$

Let $smt(R)$ be the length of a Steiner minimum tree for R in G. Then,

$$\frac{length(T)}{smt(R)} \leq \frac{1}{2}\left(\frac{mst(R)}{smt(R)} + \frac{smt_3(R)}{smt(R)}\right) \leq \frac{1}{2}\left(2 + \frac{5}{3}\right) = \frac{11}{6}.$$

4 Connected Dominating Set

An instance of Connected Dominating Set consists of a connected graph $G = (V, E)$. The objective is is to compute a minimum connected dominating set (CDS). A minimum CDS can be computed trivially if the maximum degree Δ of G is either one or two. Indeed, if $\Delta = 1$, then G contains only one edge and a minimum CDS consists of any single vertex. If $\Delta = 2$, G is a path or a cycle. When G is a path, the minimum CDS consists of all internal vertices. When G is a cycle, a minimum CDS can be obtained by deleting two adjacent vertices. So we assume $\Delta \geq 3$ from now on. We first present a generalization of the heuristic proposed by Ruan *et al.* [6].After that, we give a worst-case analysis of this heuristic which illustrates the technique of shifted submodularity.

To describe the greedy approximation, we first define a potential function. Consider an input connected graph $G = (V, E)$ and a subset $B \subseteq V$. Let $\tau(B)$ be the set of edges incident to B and $n = |V|$. Define $f_1(B) = r(\tau(B))$ where r is the rank function of the graphic matroid on G. In other words, $f_1(B)$ is equal to n minus the number of connected components in the graph $(V, \tau(B))$. Define $f_2(B)$ to be the number of connected components of the graph $G[B]$. The potential function is then defined by $f = f_1 - f_2$. Clearly, $f(\emptyset) = 0$. For any $B \neq \varnothing$, $f(B) \leq n-2$ and the equality holds iff B is a CDS. Moreover, if $n \geq 3$ then B is a CDS iff $\Delta_x f(B) = 0$ for every $x \in V$ ([6]). In addition, f_1 is submodular , but f is non-submodular.

Now let k be any positive integer parameter and consider the following greedy algorithm.

$(2k-1)$-***Greedy Algorithm***
$B \leftarrow \emptyset$;
While $\exists x \in V$ such that $\Delta_x f(B) > 0$ do
select $X \subseteq E \setminus B$ maximizing $\frac{\Delta_X f(B)}{\lvert X\rvert}$
s.t. $\lvert X\rvert \leq 2k-1$;
$B \leftarrow B \cup X$;
Output B.

Note that when $k = 1$, the $(2k-1)$-Greedy Algorithm is exactly the one presented in [6]. We will prove the following result on the performance of the $(2k-1)$-Greedy Algorithm.

Theorem 4.1. *Let B be the output by the $(2k-1)$-Greedy Algorithm. If $\Delta \geq 3$, then*

$$|B| \leq \left(1 + \frac{1}{k}\right)(1 + \ln(\Delta - 1)) \cdot opt.$$

The following "shifted submodularity" of the potential function is essential to the proof of the above theorem.

Lemma 4.1. *Let A and B be two vertex subsets. If both $G[B]$ and $G[X]$ are connected, then*

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

Proof. Since f_1 is submodular, $\Delta_X f_1(A \cup B) \leq \Delta_X f_2(A)$. Since $G[X]$ is connected, $-\Delta_X f_2(A)$ (respectively, $-\Delta_X f_2(A \cup B)$) is equal to the number of connected components of $G[A]$ (respectively, $G[A \cup B]$) adjacent to X minus one. Since $G[B]$ is connected, the number of connected components in $G[A \cup B]$ adjacent to X is at most one more than the number of connected components in $G[A \cup B]$ adjacent to X. Hence, $-\Delta_X f_2(A \cup B) \leq -\Delta_X f_2(A) + 1$. Therefore, $\Delta_X f(A \cup B) \leq \Delta_X f(A) + 1$.

Let C be a minimum CDS. The next lemma gives a lower bound on $|C|$.

Lemma 4.2. $n \leq (\Delta - 1)|C| + 2$.

Proof. We prove by induction on $|C|$ that if $G[C]$ is connected then 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.

The next lemma gives a special decomposition of C.

Lemma 4.3. *C can be decomposed into $Y_1, Y_2, \cdots, Y_p$ satisfying that*

(a) $C = Y_1 \cup Y_2 \cup \cdots \cup Y_p$,

(b) for $1 \leq i \leq p$, both $G\left[\cup_{j=1}^{i} Y_j\right]$ and $G[Y_i]$ are connected,

(c) $k+1 \leq |Y_i| \leq 2k-1$ for $1 \leq i \leq p-1$, $1 \leq |Y_p| \leq 2k-1$, and

(d) $\displaystyle\sum_{i=1}^{p} |Y_i| \leq opt + p - 1.$

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 $|T(x)|$ denote the number of vertices in $T(x)$. If T contains less than $2k$ vertices, let Y_1 consist of all vertices in T. Otherwise, there must exist a vertex $x \in C$ such that $|T(x)| \geq k+1$ and $|T(y)| \leq k$ for every child y of x. We 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 $y_1, ..., y_i$ are all children of x (see Fig. 1). 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 of all vertices in $T(y_1) \cup \cdots \cup T(y_{j+1})$ together with x and delete $Y_1 - \{x\}$ from T.

Repeating above process on the remainder of T, we will obtain a required decomposition by properly renumbering those sets in the decomposition.

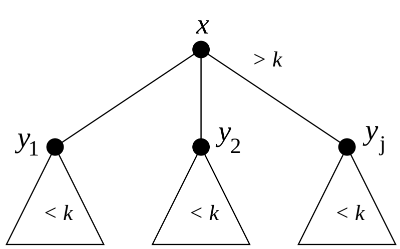

Figure 1: Case 2 in the proof of Lemma 4.3.

It's easy to show that the number p in the above lemma satisfies that $p-1 \leq (opt-1)/k$. Thus, to prove Theorem 4.1, it is sufficient to show that

$$|B| \leq (opt+p-1)\left(1+\ln(\Delta-1)\right).$$

Suppose $X_1, \cdots, X_g$ are in turn selected by the algorithm. Denote $B_0 = \emptyset$ and $B_i = X_1 \cup \cdots \cup X_i$ for $1 \leq i \leq g$. If $n \leq opt + p + \Delta - 1$, then

$$\begin{aligned}|B| &= \sum_{i=1}^{g} |X_i| \leq |X_1| + \sum_{i=2}^{g} \Delta_{X_i} f(B_{i-1}) \\ &= |X_1| + f(B) - f(X_1) \\ &\leq |X_1| + (n-2) - |X_1|(\Delta-1) \\ &= n-2-|X_1|(\Delta-2) \\ &\leq n-\Delta \leq opt+p-1 \\ &< (opt+p-1)\left(1+\ln(\Delta-1)\right).\end{aligned}$$

So we assume that $n > opt + p + \Delta - 1$. For each $0 \leq i \leq g$, let $\ell_i = f(V) - f(B_i) - (p-1)$ be the *"shifted" uncoverage* at the end of iteration i. Then

$$n-1-p = \ell_0 > \ell_1 > \cdots > \ell_g = -(p-1).$$

and $\ell_0 \geq opt + \Delta - 2 > opt$. Let $Y_1, Y_2, \cdots, Y_p$ be a decomposition of S specified in Lemma 4.3, and denote $C_j = \cup_{l=1}^{j} Y_l$ for $1 \leq j \leq p$. By Lemma 4.1,

$$\begin{aligned}f(V) - f(B_i) &= f(C) - f(B_i) \\ &= \Delta_{Y_1} f(B_i) + \sum_{j=2}^{p} \Delta_{Y_j} f(B_i \cup C_{j-1}) \\ &\leq \Delta_{Y_1} f(B_i) + \sum_{j=2}^{p} \left(\Delta_{Y_j} f(B_i) + 1\right) \\ &\leq (p-1) + \sum_{j=1}^{p} \Delta_{Y_j} f(B_i).\end{aligned}$$

Thus, $\ell_i \leq \sum_{j=1}^{p} \Delta_{Y_j} f(B_i)$ .Hence,

$$\begin{aligned}\frac{\Delta_{X_i} f(B_{i-1})}{|X_i|} &\geq \max_{1 \leq j \leq p} \frac{\Delta_{Y_j} f(B_{i-1})}{|Y_j|} \\ &\geq \frac{\sum_{j=1}^{p} \Delta_{Y_j} f(B_{i-1})}{\sum_{j=1}^{p} |Y_j|} \geq \frac{\ell_{i-1}}{opt+p-1}.\end{aligned}$$

Therefore, if $\ell_{i-1} > 0$, then

$$\frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} \geq \frac{|X_i|}{opt+p-1}.$$

In addition, for each $1 \leq i \leq g$, $\ell_{i-1} - \ell_i \geq |X_i|$. Now, let t be such that $\ell_t \geq opt > \ell_{t+1}$. Then,

$$\begin{aligned}&\sum_{i=1}^{t} |X_i| + \frac{\ell_t - opt}{\ell_t - \ell_{t+1}} |X_{t+1}| \\ &\leq (opt+p-1)\left(\sum_{i=1}^{t} \frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} + \frac{\ell_t - opt}{\ell_t}\right) \\ &\leq (opt+p-1) \ln \frac{\ell_0}{opt} \\ &\leq (opt+p-1) \ln(\Delta-1),\end{aligned}$$

and

$$
\begin{aligned}
&\frac{opt - \ell_{t+1}}{\ell_t - \ell_{t+1}} |X_{t+1}| + \sum_{i=t+2}^{g} |X_i| \\
&\le (opt - \ell_{t+1}) + \sum_{i=t+2}^{g} (\ell_{i-1} - \ell_i) \\
&= opt - \ell_g = opt + p - 1.
\end{aligned}
$$

So,

$$
|B| = \sum_{i=1}^{g} |X_i| \le (opt + p - 1)\left(1 + \ln(\Delta - 1)\right).
$$

References

[1] S.-K. Chang, The generation of minimal trees with a Steiner topology, *J. ACM*, 19 (1972) 699-711.

[2] S.-K. Chang, The design of network configurations with linear or piecewise linear cost functions, *Symp. on Computer-Communications, Networks, and Teletraffic*, (1972) 363-369.

[3] U. Feige, A threshold of $\ln n$ for approximating set cover. *Proc. the Twenty-Eighth Annual ACM Symp. Theory of Computing* (1996) 314–318.

[4] T. Fujito, On approximability of the independent/connected edge dominating set problems, *Lecture Notes in Computer Sciences*, vol. 1974, Springer, Berlin, 2000, pp. 117–126.

[5] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*, Wiley, 1999.

[6] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K. Ko. A Greedy Approximation for Minimum Connected Dominating Sets. Theoretical Computer Science, 329 (2004) 325-330.

[7] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* 4(2) (1982) 385–393.

[8] A.Z. Zelikovsky, The 11/6-approximation algorithm for the Steiner problem on networks, *Algorithmica* 9 (1993) 463-470.