Network Design Foundation
Fall 2015
Connected Dominating Sets with Local Information

Instructor: Mohammad T. Hajiaghayi
Saba Ahmadi, Sheng Yang

December 10, 2015

1 Introduction

Here we consider approximation algorithms for the Connected Dominating problem using both global and local information.

A dominating set is one in which each vertex is either in the dominating set or adjacent some vertex in the dominating set. The Connected Dominating Set problem is defined as follows.

Definition 1 (Connected Dominating Set (CDS)) *Find a minimum size subset S of vertices, such that the subgraph induced by S is connected and S forms a dominating set.*

This problem is known to be NP-hard, so we can only approximate it. In this note, we focus on the setting of local information, which is,

Definition 2 (r-hop of local information) *If we know a subgraph S' of S, then with r-hop local information, we know the induced graph of $\bar{S} = \{v|d(v, S') \leq r\}$, and the degree of all nodes in $\bar{S}$.*

This note is organized as follows. We present two approximation algorithms for this problem. In section 2, we talk about a naive greedy algorithm which gives a Δ-approximation, where Δ is the largest degree in a graph. Then with some modification, we explain a $2(1 + H(\Delta))$-approximation which needs 2-hop of information in section 3. After that in section 4, the result was improved. We still have $2(1 + H(\Delta))$-approximation but needs only 1-hop of information. Finally, in section 5, comes the original part, where we improved the approximation ratio to $H(\Delta) + 2\sqrt{H(\Delta)} + 1$.

2 Algorithm 1

We introduce an algorithm with 1-hop local information (by [1]) that finds a connected dominating set by growing a tree. This algorithm will not give a good approximation ratio (it gives Δ approximation ratio), but will shed some light on how to construct one with good performance.

To help understand everything, we color all the vertices with black, gray and white. Black vertices are those selected in our solution. Gray vertices are not selected, but dominated by the black vertices. The rest are white. Initially, all vertices are colored white (unmarked). When we scan a vertex, we color it black and color all its white neighbors gray (mark these nodes as dominated). We start from an arbitrary initial node. In the following steps, we find a gray vertex which has the maximum number of white numbers.

The algorithm continues until all vertices are black or gray. The set of black vertices forms a dominating set. With the nature of the algorithm, we know this dominating set is connected, so we get a connected dominating set.

This algorithm is simple and intuitive, but cannot give good approximation. The approximation ratio can be as bad as Δ. Consider the following example (Fig 1).

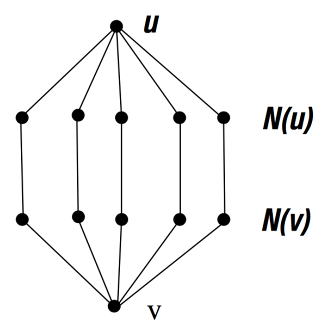

Figure 1: The Scanning Rule Fails.

We start from u. Then all of its neighbors $N(u) = \{v|(u,v) \in E\}$ will be colored gray. In the next step we will select a vertex u' from $N(u)$ and color its neighbor gray. Now that all gray vertices has exactly one white neighbor. We could possibly pick a vertex from $N(u)$ again. This might continues until all the vertices from $N(u)$ have been selected. After that, a vertex from $N(v)$ will be chosen and v will be colored. Thus our solution uses $d+2$ vertices. But optimal solution will pick a CDS of size 4, which is a path from u to v.

3 Algorithm 2

We can modify the selecting method to achieve a better approximation ratio. We define a new operation of selecting a pair of adjacent vertices u and v, such that u is grey and v is white. The white neighbors of this pair, is the set of white vertices that are adjacent to at least one of u and v.

Algorithm 1: CDS with 2-hop Information

```
Data: Graph G = (V, E)
Result: A connected dominating set S
S ← an initial point s;
while S is not a dominating set do
    S̄ ← the node or the pair that maximize the number
      of white neighbors;
    S ← S ∪ S̄;
```

At each step, we will either select a single vertex, or a pair of vertices, whichever has greater number of white neighbors. In another word, we require 2-hop of information. This improves the approximation ratio to $2(1 + H(\Delta))$. Let OPT_{DS} be the set of vertices in an optimal dominating set, we have the following theorem.

Theorem 1 *We can find a connected dominating set of size at most* $2(1 + H(\Delta)) \cdot |OPT_{DS}|$ *with 2-hop local information.*

Proof: Suppose the optimal solution is OPT, we partition the set of all vertices V into $\{S_1, \ldots, S_{OPT}\}$, such that S_i includes all the nodes covered by a certain node v_{OPT_i} (including itself). Ties are broken arbitrarily.

The proof will be based on a charging scheme. Each time we scan a vertex and add it to the our solution, we will mark some vertices. If we select a single vertex and color x vertices gray, we will charge each of them $1/x$. If we select a pair of adjacent vertices and color x vertices gray, we charge each of them $2/x$. Note a vertex is charged only when it turns from white to gray, so we only charge it once. Here the sum of charges is equal to number of vertices in our solution.

We now give an upper bound on the total charges assigned to vertices in S_i. Let u_j be the number of unmarked vertices in S_i after step j. For simplicity, let us assume that at each step some vertices of S_i are marked, so the number of unmarked vertices decreases at each step.

The number of unmarked vertices after the first step is $u_0 - u_1$ and each of them will be charged at most $\frac{2}{u_0 - u_1}$. Once a vertex from S_i is marked, vertex i becomes an eligible vertex to be scanned since it's neighbour of a marked vertex. In the j^{th} step, the number of vertices of set S_i that get marked is $u_j - u_{j+1}$ and the cost to assigned to each vertex is at most $\frac{2}{j}$ as vertex i was an eligible vertex to be scanned. Let $u_k = 0$. Adding up all the charges we get:

$$\frac{2}{u_0 - u_1}(u_0 - u_1) + \sum_{j=1}^{k-1} \frac{2}{u_j}(u_j - u_{j+1})$$
$$\leq 2 + 2\sum_{j=1}^{k-1} \frac{u_j - u_{j+1}}{u_j}$$

One can show that this is at most $2(1 + H(\Delta))$.

Every vertex is dominated by some vertex in OPT. So every node exists in some S_i, such that $1 \leq i \leq |OPT|$. Sum of charges assigned to vertices in S_i is at most $2(1 + H(\Delta))$. Putting it together, total charge assigned to all vertices is at most $2(1 + H(\Delta))|OPT|$. Thus our algorithm produces a $2(1 + H(\Delta))$ approximation. ■

4 Local 1-hop algorithm for CDS

4.1 1-hop information algorithm

Every time, we choose one node that maximize the number of newly covered nodes, and then choose one of the newly covered nodes uniformly randomly.

Algorithm 2: CDS with 1-hop Information

```
Data: Graph G = (V, E)
Result: A connected dominating set S
1 S ← an initial point s;
2 while S is not a dominating set do
3     v ← the node that maximize the number of newly
      covered nodes;
4     u ← an uniformly randomly chosen node from the
      newly covered nodes;
5     S ← S ∪ {u, v};
```

4.2 Analysis

We stick to the charging scheme. Whenever we select a node in line 3 in the previous algorithm, we put a charge. The charge is spread uniformly among all the newly covered nodes. For a node select in line 4, we do not charge it. Thus, the number of nodes chosen is twice the total charge.

Suppose the optimal solution is OPT, we partition the set of all vertices V into $\{S_1, \ldots, S_{OPT}\}$, such that S_i includes all the nodes covered by a certain node v_{OPT_i} (including itself). Ties are broken arbitrarily. Then what we need to do is to bound the total charge in each S_i.

To help explain the problem, we color all the nodes. At first, all nodes are colored white. Then, the starting point is colored black, and all its neighbors

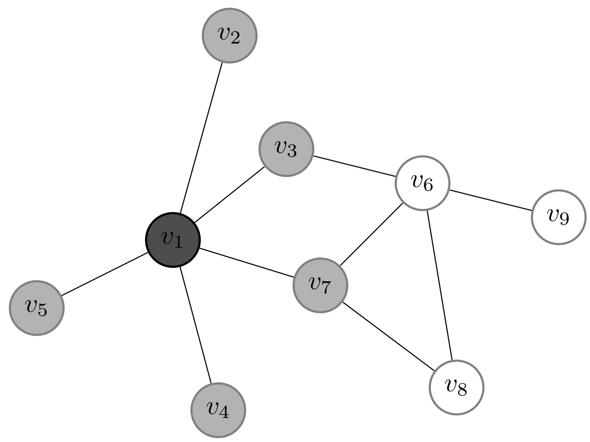

Figure 2: An example of coloring

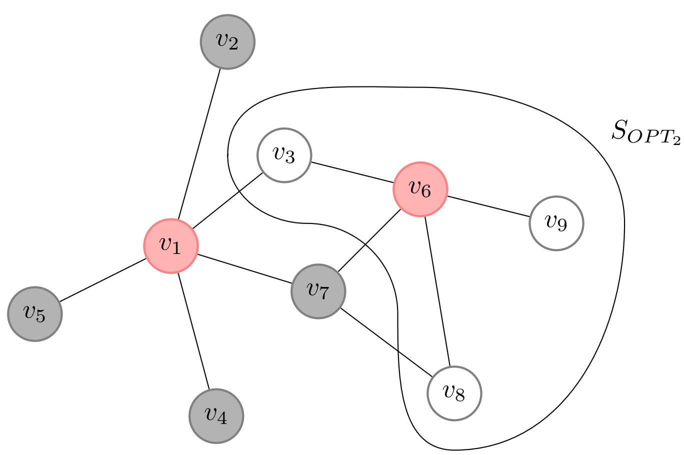

Figure 3: An example of S_i partition

are colored gray. After that, whenever we pick a node, we color it black, and all the nodes dominated by it is colored gray.

We split the charging process into two phases. The first phase ends when a node in S_i is chosen, or marked black. In the second phase, the vertex v_{OPT_i} is available, but we choose some other nodes.

We first bound the charge in the second phase. It looks like the analysis for set cover, and gets similar answer: $H(\Delta)$, where Δ is the largest degree in the graph. Suppose the number of non-dominated nodes in S_i after each round is $k_0, k_1, \ldots, k_t, k_{t+1} = 0$, then if we choose node u at step j instead of v_{OPT_i}, we can guarantee that the number of newly covered nodes is lower bounded by k_j, so the charge on each newly covered node is upper bounded by $\frac{1}{k_j}$. This means the total charge in S_i in phase 2 during step j is upper bounded by

$$\frac{k_j - k_{j+1}}{k_j} \leq \sum_{s=k_{j+1}-1}^{k_j} \frac{1}{s}$$

Thus the total charge in S_i in phase 2 is bounded by

$$\begin{aligned}\sum_{r=0}^{t} \frac{k_r - k_{r+1}}{k_r} &\leq \sum_{r=0}^{t} \sum_{s=k_{j+1}-1}^{k_j} \frac{1}{s} \\ &\leq \sum_{s=1}^{k_0} \frac{1}{s} \leq \sum_{s=1}^{|S_i|-1} \frac{1}{s} \\ &= H(|S_i| - 1) \leq H(\Delta)\end{aligned}$$

Where H is harmonic function.

As for the first part, we notice the following fact: whenever a charge p comes into S_i, there is probability p that phase 1 will end instantly. This comes from the algorithm directly, where both charge and choice are uniformly random.

Think of the following problem: for $1 \leq i \leq n$, let X_i be a Bernoulli random variable with expected value $p_i \in [0, 1]$. Let T be the random variable denoting the smallest i such that $X_i = 1$(or n if $X_i = 0$ for all I). Then $\mathbb{E}_T[\sum_{i=1}^{T} p_i] \leq 1$. This is done by induction. It is trivial if $n = 1$. For $n > 1$,

$$\mathbb{E}_T\left[\sum_{i=1}^{T} p_i\right] = p_1 + (1 - p_1)\mathbb{E}_T\left[\sum_{i=2}^{T} p_i | X_1 = 0\right] \leq p_1 + (1 - p_1) \cdot 1 = 1$$

where the inequality come from inductive hypothesis[2]. Putting everything together, we have an algorithm that use 1-hop information, and achieves approximation ratio $2(H(\Delta) + 1)$.

5 We can even do better

It is easy to see that there is a gap of 2 between global and local algorithm. This existence makes sense, but can we do better?

We can get better, and our improved approximation ratio is

$$H(\Delta) + 2\sqrt{H(\Delta)} + 1$$

References

[1] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[2] C. Borgs, M. Brautbar, J. Chayes, S. Khanna, and B. Lucier, "The power of local information in social networks," in *Internet and Network Economics*. Springer, 2012, pp. 406–419.