Noname manuscript No.
(will be inserted by the editor)

Revisiting Connected Dominating Sets: An Almost Optimal Local Information Algorithm*

Samir Khuller · Sheng Yang

the date of receipt and acceptance should be inserted later

Abstract In this paper we consider the classical Connected Dominating Set (CDS) problem. Twenty years ago, Guha and Khuller developed two algorithms for this problem - a centralized greedy approach with an approximation guarantee of $H(\Delta)+2$, and a local information greedy approach with an approximation guarantee of $2(H(\Delta)+1)$ (where $H()$ is the harmonic function, and Δ is the maximum degree in the graph). A local information greedy algorithm uses significantly less knowledge about the graph, and can be useful in a variety of contexts. However, a fundamental question remained - can we get a local information greedy algorithm with the same performance guarantee as the global greedy algorithm without the penalty of the multiplicative factor of "2" in the approximation factor? In this paper, we answer that question in the affirmative.

1 Introduction

A connected dominating set (CDS) in a graph is a subset of vertices that induces a connected subgraph, and is simultaneously a dominating set. A dominating set is a subset of vertices such that every node in the graph, is either in the dominating set, or adjacent to a node in the dominating set. Finding a minimum connected dominating set is NP-hard, and thus for the last twenty years, researchers have explored approximation algorithms for this problem starting with the work of Guha and Khuller [7][1]. One of the

* This work is supported by NSF grant CCF 1217890 and CCF 1655073 (Eager). A preliminary version of this work was published in APPROX-RANDOM 2016.

Samir Khuller, Sheng Yang
Dept. of Computer Science
University of Maryland, College Park, USA
E-mail: {samir,styang}@cs.umd.edu

[1] This work was recently awarded the ESA Test of Time Award.

original motivations for the problem was in building a backbone for routing in the context of wireless ad hoc networks. Over time many other applications have been explored in the context of social networks and AI [2, 11]. See other applications and results in the book by Du and Wan [4].

In their paper, Guha and Khuller [7] developed two simple approaches - the first one is a "local" approach, where we start from a single vertex in the solution, and incrementally (greedily) add neighboring nodes while maintaining a connected subset of nodes at all times. It is tempting to imagine that adding one node at a time, maintaining connectivity might work well. However, this does not work and there are instances where we might end up with a solution with $\Omega(n)$ nodes, while the optimum solution has only $O(1)$ nodes! Interestingly, a simple modification of this algorithm that explores a 2-hop neighborhood making greedy choices at each step that involves selecting *up to two* nodes at each step, works much better and they show that a $2(H(\Delta)+1)$ approximation can be obtained for this problem ($H(n) = \sum_{i=1}^{n} \frac{1}{i}$ is the harmonic function, and Δ is the largest degree in the graph). The main benefits of the local information algorithm are that no knowledge of the entire graph is needed at each step, and only the part of the "explored" graph suffices to select the next node. This may not be a useful for a static graph, as you can calculate it once and forever. To get a CDS for a graph in a social network setting, where the level of information about the network revealed is limited to the neighbors of explored nodes, motivates the need for local information algorithms.

They also developed an improved global algorithm that is again a greedy algorithm, and constructs a solution that does not maintain any connectivity property, and is only connected at the very end. This centralized greedy algorithm yields a bound of $H(\Delta) + O(1)$, eliminating the factor of 2, and gets us close to the lower bound on this problem of $(1-\epsilon)H(\Delta)$ due to set cover hardness [6]. Despite much interest (see book by Du and Wan [4]), the key question remained open for two decades - is there a local information algorithm that gives us the same bound as the global algorithm?

In this paper, we answer this question in the affirmative. We develop a very simple local information algorithm, and are able to show that it matches the bound of the global algorithm. Such a result is especially surprising due to its simplicity.

Connected Dominating Sets became a central topic in the context of wireless ad hoc networks, where the CDS acts as a routing backbone for packet routing. Often it is expensive to connect all the nodes, as the cost can become prohibitive, and in this case it is fine to connect *most* of the nodes (or a given fraction). Liu and Liang [10] formalized the problem of *partial* connected dominating sets (PCDS) and provided heuristics without performance guarantees. Avrachenkov et al. [1] defined the budgeted connected dominating set problem (BCDS) where we have a budget of k nodes and we wish to find a connected set of at most k nodes that maximizes the number of nodes it can cover. Inspired by applications in social networks, they developed practical heuristics using only local information. Khuller et al. [9] developed the first algorithms with

theoretical guarantees for both these problems with approximation factors of $O(\ln \Delta)$ for PCDS and $\frac{1}{13}\left(1 - \frac{1}{e}\right)$ for BCDS.

Extensions to node weighted versions were considered by Guha and Khuller [8] as well. Extensive research was subsequently done on this topic with the development of distributed algorithms [5], as well as for many special classes of graphs [4].

1.1 Our Contributions

Our results can be summarized as follows

- In Section 3, for the Connected Dominating Set problem, we obtain the first local information algorithm whose approximation ratio is within additive constant to global algorithm, i.e. $H(\Delta) + O(1)$. To be precise, our approximation guarantee is $H(2\Delta + 1) + 2$. This algorithm requires 2-hop local information (see Section 2 for definition).
- In Section 4, with 1-hop local information, we obtain a randomized $H(\Delta) + 2\sqrt{H(\Delta)} + 1$ approximation algorithm. In addition to better approximation ratio, it also runs faster than the CDS algorithm proposed by Guha and Khuller [7] (Section 2.3), because it explores fewer nodes.

2 Background

We first review existing approaches [7, 2].

2.1 Global Algorithm for CDS

The global algorithm (of [7]) runs in two phases. Initially, all nodes are colored white. In the first phase, the algorithm iteratively adds a node to the solution, colors it black and all its adjacent white nodes gray. A *piece* is defined as a white node or a black connected component. A new node is chosen to be colored black to get maximum reduction in the number of pieces. This phase ends when no such node exists that can give non-zero reductions. At this time, there are no white nodes left. Black nodes are selected nodes, gray nodes are nodes that are dominated, i.e. adjacent to black nodes.

In the second phase, we start with a dominating set that consists of several black components that we need to connect. The connection is done by recursively connecting pairs of black components with a chain of vertices, until there is only one black component, which will be our final solution.

The approximation ratio for this algorithm is $H(\Delta) + 2$, where $H(n)$ is the harmonic function.

2.2 2-hop Local Information Algorithm for CDS

The same paper [7] proposed another algorithm, using only local information. Instead of using information about the entire graph, it only relies on information within 2-hops to the nodes chosen in the solution. The formal definition of local information is as follows.

Before we define what local information is, we first define the distance between a node and a set of nodes.

Definition 1 (Distance) In an undirected graph, denote the distance between u and v in a graph as $d(u, v)$. It is the length of shortest path from u to v. $d(u, S)$ is defined to be $\min_{v \in S} d(u, v)$.

We now define the local neighborhood of some node, or a set of nodes.

Definition 2 (Local Neighborhood) Given a set of nodes S in graph G, the r-hop neighborhood around S is the induced subgraph of G containing all nodes v such that $d(v, S) \leq r$. We denote the r-hop neighborhood as $G[N^r(S)]$, where $N^r(S) = \{v | d(v, S) \leq r\}$.

An algorithm with local information uses information only within the local neighborhood of the nodes it has chosen. To be specific, if at some step, the set of nodes that an algorithm has chosen is S, and we have r-hop local information, then we know the induced graph of $N^r(S)$, as well as the degree of nodes in $N^r(S)/N^{r-1}(S)$.

From an arbitrary starting node, nodes are added iteratively. For each loop in this algorithm, one chooses a node, or a node and one of its neighbors. This means we need knowledge of 2-hop neighborhood to maximize the number of newly covered nodes, which explains why it uses 2-hops of local information.

Algorithm 1: CDS with 2-hop Local Information

Data: Graph $G = (V, E)$
Result: A connected dominating set S

```
s ← an arbitrary node;
S ← {s};
while S is not a dominating set do
    S̄ ← a node u in N¹(S) or a node u in N¹(S) and
      one of its neighbors in N¹(u) (which also lies in
      N²(S)) that maximize the number of newly
      covered nodes;
    S ← S ∪ S̄;
```

The approximation ratio for this algorithm is $2(H(\Delta) + 1)$. We can improve it in practice by maximizing the number of newly covered nodes for each new node selected, which is used in [11], but the theoretical worst case bound is the same.

2.3 1-hop Local Information Algorithm for CDS

Borgs et al. [2] proposed a randomized 1-hop local algorithm, which is based on the one described in Section 2.2. We use the same coloring, i.e. black for selected vertices, gray for unselected but covered ones, and white for uncovered vertices. Instead of choosing a node or a pair of adjacent nodes greedily, it chooses only one gray node to maximize the number of newly covered nodes, and in addition selects one of the newly covered nodes uniformly at random. The maximization process only requires 1-hop neighborhood information, and we do not worry about the random node we are about to choose. Not only does this algorithm require less information, it runs much faster. The approximation ratio is unchanged, and is $2(H(\Delta) + 1)$.

Algorithm 2: Randomized CDS with 1-hop Local Information

Data: Graph $G = (V, E)$
Result: A connected dominating set S
$s \leftarrow$ an arbitrary node;
$S \leftarrow \{s\}$;
while S *is not a dominating set* **do**
  $v \leftarrow$ a node in $N^1(S)$ that maximize the number of newly covered nodes;
  $u \leftarrow$ a uniformly randomly chosen node from $N^1(v) - N^1(S)$, i.e. the newly covered nodes;
  $S \leftarrow S \cup \{u, v\}$;

2.4 Inspiration

Comparing the two algorithms using global information and local information, we find a gap of factor 2 in the approximation ratio: $2(H(\Delta)) + O(1)$ for a local information algorithm versus $H(\Delta) + O(1)$ for global algorithm. This looks reasonable: perhaps we are always better off when offered more information. But is it really the case? Is this gap the price we paid for lack of information essential, or the same approximation ratio can be achieved regardless of limited information. Surprisingly, our answer is that with local information, we can still get $H(\Delta)+O(1)$ approximation. To be specific, an $H(2\Delta+1)+1$ approximation.

3 Improved 2-Hop Local Information Algorithm

3.1 Algorithm

S is initially a set that consists of a single node. We iteratively add nodes within $N^2(S)$ to S. When new nodes are added to S, the 2-hop neighborhood of S expands, and we repeat this process. By the end of the process, we have

colored all the nodes. Initially, all nodes are white. When a node is added to S, we color it black, and all its white neighbors gray. The selected black nodes will form components, and these components may merge when additional nodes are selected.

At each step, we look within a 2-hop neighborhood of S, i.e., among the nodes that are either gray or adjacent to gray nodes. Add the node v that maximizes $2w_v + c_v - 1$, where c_v is the number of different black components connected to v, and w_v is the number of white nodes in $N^1(v)$. Our knowledge of the graph is enriched when we add nodes to our solution. The algorithm ends when there is no v such that $2w_v + c_v - 1 > 0$. Note that if v was gray before the selection, c_v is guaranteed to be greater or equal to 1. If it was white, $c_v = 0$.

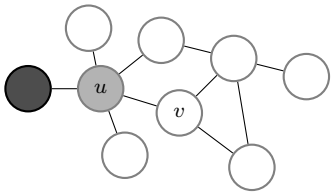

Fig. 1 Consider u and v as potential choices. For node u, $c_u = 1$, as there is only one adjacent component. Note that $w_u = 4$, since u has four white neighbors, and itself is not white, hence $2 \cdot w_u + c_u - 1 = 8$. For node v, since it is white, there is no adjacent black component, so $c_v = 0$. Note w_v equals 3, since node v itself and two of its neighbors are white. So the value is $2 \cdot w_v + c_v - 1 = 6 + 0 - 1 = 5$.

For completeness, here is the pseudo code.

Algorithm 3: CDS with 2-hop Local Information

Data: Graph $G = (V, E)$
Result: A connected dominating set S

$s \leftarrow$ an arbitrary node;
$S \leftarrow \{s\}$;
$\bar{V} \leftarrow N^2(S)$;
while $\exists v \in \bar{V}$, *s.t.* $2w_v + c_v - 1 > 0$ **do**
  $v \leftarrow \text{argmax}_{v \in \bar{V}} 2w_v + c_v - 1$;
  $S \leftarrow S \cup \{v\}$;
  $\bar{V} \leftarrow N^2(S)$;
  for all nodes in V, update c_v, w_v;

3.1.1 Correctness

First we prove that what we get is a dominating set. If not, then there exists a node that is not dominated, i.e. a white node u. But $2w_u + c_u - 1 = 2w_u + 0 - 1 \geq 2 - 1 > 0$, we would have added this node to our solution, a contradiction.

Next, we prove the following theorem that this dominating set is indeed connected.

Theorem 1 *The solution returned by Algorithm 3 is connected.*

We have the following observation:

Observation 2 *When a node is added to our solution, it is within 2-hops of some black node.*

It is true because otherwise, this node would be beyond our knowledge and would not be considered at this step. This ensures the following corollary:

Corollary 1 *Each newly added node can be connected to existing components at the cost of at most one additional node.*

Proof (Proof of Theorem 1) We prove this by contradiction. Suppose the algorithm gives a dominating set that is not connected, i.e. has more than one component. Exactly one of these components will contain the starting node, call it starting component. Consider the first time when a node u outside the starting component joins our solution. According to Corollary 1, there must exist another node v that can join u to the starting component. Since u is disconnected from the starting component, v is not in our solution. But at the end of the algorithm, since $c_v \geq 2$, $2 \cdot w_v + c_v - 1 \geq 0 + 2 - 1 > 0$, which means that v can be added to our algorithm. This gives a contradiction. □

So our algorithm will indeed give a connected dominating set.

3.2 Analysis

We are going to charge the cost of each selected node and bound the total charge, which in turn bounds the number of nodes we select. One node will be charged once, and all other nodes will be charged twice in the end. Based on the number of times a node is charged at a certain time, we define a single-charged black node (we will ensure only black nodes can be single-charged) and a fully-charged node:

Definition 3 A single-charged black node is a node that was charged once when it turned directly from white to black, and has not been charged afterwards.

Definition 4 A fully-charged node is a node that got charged twice.

We state without proof of the following observation, which will become obvious when the charging scheme is clear.

Observation 3 *All white nodes are uncharged. All gray nodes are fully charged. Black nodes are either single-charged or fully-charged.*

To make the charging scheme valid, we need the following lemma for single-charged black nodes, which will be proved after the charging scheme is described.

Lemma 1 *Each black component contains exactly one single-charged black node.*

Now we describe the charging scheme. In the beginning, all nodes are white and uncharged. Each time a node is selected to be added to the solution, we distribute the charge for the node into shares. By adding all the shares, we can eventually bounded the total cost. Recall we always select a node v that maximizes $2w_v + c_v - 1$, where w_v is the number of white nodes in $N^1(v)$, and c_v is the number of black components adjacent to v. For selecting v, we distribute a charge of one into $2w_v + c_v - 1$ shares which will possibly be charged to v, some of its neighbors, and even some remote singled-charged nodes.

Node v may be white or gray before the selection. If it was white, it means that v is not adjacent to any black nodes, so $c_v = 0$. We charge v itself one share, i.e. $1/(2w_v + c_v - 1) = 1/(2w_v - 1)$, and turn it black. At this time, v is considered a single-charged black node as it turns directly from white to black, and is only charged once. The remaining $2(w_v - 1)$ shares are split uniformly among all its $w_v - 1$ white neighbors (notice w_v also counts for v itself), each taking 2 shares, i.e. $2/(2w_v - 1)$. These neighbors are now gray since they are now adjacent to some black node v. We consider these gray nodes to have been charged twice, one for each share, which means they are fully charged and will not be charged any more, even if they turn black at some later step.

If v was gray, we cannot charge it as it is already fully charged. Instead, we charge its white neighbors and neighboring black components. Every white neighbor of v will take two shares, the same as previous case (there are w_v of them as v is not white). For all but one adjacent component, a share of charge is placed on the single-charged black node of this component (assuming the correctness of Lemma 1). So the number of shares is $2w_v+(c_v-1) = 2w_v+c_v-1$, the same as the number of shares we split. A visual explanation is in Figure 2.

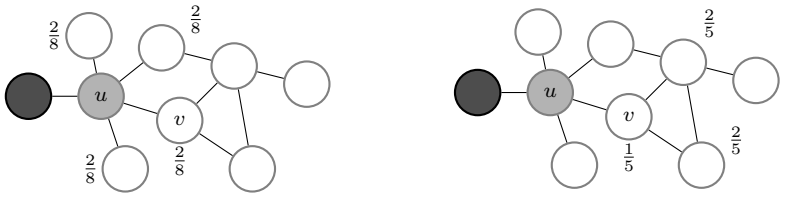

Fig. 2 Consider the charging for u and v if they were selected at this step. For u, $2 \cdot w_u + c_u - 1 = 8$, each share is $1/8$, and each white neighbor of u gets 2 shares. For node v, $2 \cdot w_v + c_v - 1 = 6 + 0 - 1 = 5$, each share is $1/5$. Each white neighbor gets 2 shares, but since node v goes from white directly to black, it only gets one share, and becomes a single-charged black node.

We now prove the correctness of Lemma 1.

Proof (Proof of Lemma 1) A single-charged black node comes into existence when a white node is chosen and added to our solution. According to the charging scheme, this node itself forms a component, and it has exactly one single-charged black node.

Components are connected when a gray node is chosen which connects several components. Since all but one component was charged, assuming all existing components have exactly one single-charged black node, the resulting component also has exactly one single-charged black node. A visual explanation is in Figure 3. □

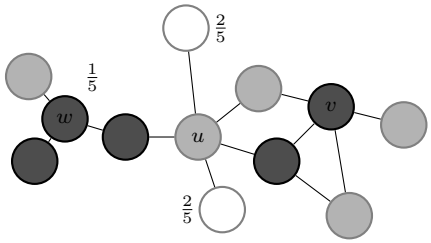

Fig. 3 Suppose the black nodes are already chosen, and we are adding node u to the solution. There are two black components adjacent to u, so $c_u = 2$. Thus $2w_u + c_u - 1 = 4 + 2 - 1 = 5$. Each white neighbor of u gets 2 shares. For the two components, all but one component will get a share. This share is charged against the single-charged black node (node w for the left component and node v for the right component) of the component, which may not be the node adjacent to u. After charging, the two components are merged, and the merged component still has exactly one single-charged black node, i.e. node v. Note there is no charge against node u.

Everything prepared, we state the main theorem and prove it by bounding the total charge.

Theorem 4 (Main Theorem for 2-hop Local Information Algorithm) *The improved 2-hop local information greedy algorithm gives $H(2\Delta + 1) + 2$ approximation.*

Proof Suppose the optimal solution is OPT, $|\text{OPT}| = k$, with vertices v_{OPT_1}, v_{OPT_2}, ..., v_{OPT_k}. We partition all nodes into S_{OPT_1}, S_{OPT_2}, ..., S_{OPT_k}. Without loss of generality, we reorder the nodes, and use i to denote vertex v_{OPT_i}. So S_i contains vertex i and its neighbors. Ties are broken arbitrarily as long as $i \in S_i$.

We only consider rounds in which the charge on S_i changes. To describe the total number of charges that nodes in S_i can receive, we define p_i^j as the following value for S_i in step j:

$$2 \cdot w_i^j + b_i^j - 1$$

where w_i^j is the number of white nodes in S_i at step j, b_i^j is the number of single-charged black nodes at step j in S_i. Recall c_i is defined earlier as the

number of different black components connected to vertex i. A symbol (e.g. w_i, b_i, c_i) with a superscript j, i.e. w_i^j, b_i^j, c_i^j, denotes the corresponding value at step j. Then $p_i^1 \leq 2\delta(i) + 1$ ($\delta(i)$ is the degree of node i). A white node can receive two charges (in fact all white nodes will receive two charges, except for one, which is the only single-charged black node when the algorithm ends). Either it will receive two charges when it first becomes gray, and never receive any more charge. Or it receives one charge when it becomes a single-charged black node, and receives another one to become a charged black node. We have the following lemma:

Lemma 2 *If the total charge inside S_i changes at step j, then $p_i^j - p_i^{j+1} \geq 1$.*

Proof Total charge changes when at least one node in S_i receives charges. There can be three cases: this node was white, gray, or black. If this node was white before charging, either it is now gray, which means it receives two charges, or it is now black, meaning one charge. Therefore this node will cause p_i^j to decrease by 1 or 2 (or more since there can be more than one such node, p_i^j may decreases more than that). If it was gray, it must have been fully charged and will not receive any more charge. If it was black, it must have been a single-charged black node to receives one charge. In this case, p_i^j also decrease one. For all three cases, either there is no charge change in S_i, or there is a decrease on p_i^j of at least 1. □

We use p^j instead of p_i^j when there is no confusion. Notice $b_i^j \leq c_i^j$, since every single-charged black node corresponds to a component, but the single-charged black node may not be in S_i. Consider each step before i is selected (if i is selected, then the only possible charge S_i will ever get is one share for a single-charged black node, which will be handled separately with the case $p_i^j \leq 0$). When $j = 1$, it is the first time some vertex in S_i got charged, the total charge is bounded by 1. For all other steps, suppose the selected node is v. Since some node in S_i has been charged, it is 1-hop away from some black node, and i is 2-hop away. This implies i is also a valid choice. We have

$$2w_v^j + c_v^j - 1 \geq 2w_i^j + c_i^j - 1 \geq 2w_i^j + b_i^j - 1 = p^j.$$

The number of shares that S_i receives at step j is $p^j - p^{j+1}$, so the total charge in this step is:

$$\frac{p^j - p^{j+1}}{2w_v^j + c_v^j - 1} \leq \frac{p^j - p^{j+1}}{p^j}.$$

This holds when $2w_i^j + c_i^j - 1 > 0$, which is guaranteed to be true when $p_i^j > 0$. The inequality will break down if $p_i^j \leq 0$, e.g. when $w_v^j = 0$, $c_v^j = 1$, and $2w_i^j + c_i^j - 1 = 0$. To fix it, we notice $\forall S_i, \exists k_i, s.t.\ p_i^{k_i} > 0$ and $\forall t > k_i, p_i^t \leq 0$. Thus we can take out the last term and bound it separately. So the total charge until step k_i (including step k_i) is upper bounded by:

$$1 + \sum_{j=2}^{k_i} \frac{p^j - p^{j+1}}{p^j} = 1 + \sum_{j=2}^{k_i} \sum_{t=p^{j+1}+1}^{p^j} \frac{1}{p^j}$$

$$
\begin{aligned}
&= 1 + \sum_{j=2}^{k_i-1} \sum_{t=p^{j+1}+1}^{p^j} \frac{1}{p^j} + \sum_{t=p^{k_i+1}+1}^{p^{k_i}} \frac{1}{p^{k_i}} \\
&\leq 1 + \sum_{j=2}^{k_i-1} \sum_{t=p^{j+1}+1}^{p^j} \frac{1}{t} + \left(\sum_{t=p^{k_i+1}+1}^{\max\{p^{k_i+1}+1,1\}-1} 1 + \sum_{t=\max\{p^{k_i+1}+1,1\}}^{p^{k_i}} \frac{1}{t} \right) \\
&\leq 1 + \left(\sum_{j=2}^{k_i-1} \sum_{t=p^{j+1}+1}^{p^j} \frac{1}{t} + \sum_{t=\max\{p^{k_i+1}+1,1\}}^{p^{k_i}} \frac{1}{t} \right) + \sum_{t=p^{k_i+1}+1}^{\max\{p^{k_i+1}+1,1\}-1} 1 \\
&= 1 + \sum_{j=2}^{p^1} \frac{1}{j} + \sum_{p^{k_i+1}+1}^{\max\{p^{k_i+1}+1,1\}-1} 1 \\
&= 1 + H(p^1) + (0 - p_i^{k_i+1})
\end{aligned}
$$

where $H(n) = \sum_{i=1}^{n} \frac{1}{i}$ is harmonic sum. The last equality uses the fact that $2w_v + c_v - 1 \geq -1$.

Using $p_i^{k_i+1} \leq 0$, $p_i^t \geq -1$, combined with our assumption that p_i^j decreases at each step, there is at most one more step before the whole algorithm stops. So the total amount of charge for this step is upper bounded by:

$$(p_i^{k_i+1} - p_i^{k_i+2}) \cdot 1 \leq (p_i^{k_i+1} + 1).$$

Adding together, we have,

$$1 + H(p^1) + (0 - p_i^{k_i+1}) + (p_i^{k_i+1} + 1) = H(p^1) + 2.$$

As $p^1 = 2w_i + b_i - 1 = 2w_i - 1 \leq 2\Delta + 1$, the total charge in S_i is bounded by $H(2\Delta + 1) + 2$. Since there are $|OPT|$ different S_i, the total charge is bounded by $|OPT|(H(2\Delta + 1) + 2)$, which is also the upper bound for the number of nodes we choose. □

4 Improved 1-Hop Local Information Algorithm

4.1 Intuition

Recall the algorithm by Borgs et al. [2], which selects a random white neighbor when a gray node is chosen. It is too expensive to select a random node every time. Instead of picking a random neighbor every time, we can pick it with some probability p. The total approximation ratio will be (this is only an intuition, detailed proof is in Section 4.3) $(1+p)(\frac{1}{p} + H(\Delta))$ that can be minimized when $p = \frac{1}{\sqrt{H(\Delta)}}$. This works given the assumption that Δ is known before hand, which is not always the case.

4.2 Algorithm

Instead of using the largest degree in the graph, every time when calculating p, we use the largest degree in the explored graph. In other words, the largest degree in $N^1(S)$. Below is the pseudocode.

Algorithm 4: Improved Algorithm for CDS with 1-hop Local Information

Data: Graph $G = (V, E)$
Result: A connected dominating set S

```
1  s ← an arbitrary node;
2  S ← {s};
3  while S is not a dominating set do
4      d ← the largest degree in N^1(S);
5      p ← 1/√H(d);
6      v ← a node in N^1(S) that maximizes the number of
         newly covered nodes;
7      S ← S ∪ {v};
8      begin with probability p
9          u ← a node chosen from the newly covered
             nodes uniformly at random;
10         S ← S ∪ {u};
```

4.3 Analysis

As in the previous analysis, we do charging, and partition all the nodes into $S_1, S_2, \ldots, S_{OPT}$ in the same manner. We bound the total charge inside S_i, and the total number of nodes chosen in our solution that corresponds to these charges.

Theorem 5 (Main Theorem for 1-hop Local Information Algorithm) *The improved 1-hop local information greedy algorithm gives $H(\Delta)+2\sqrt{H(\Delta)}+1$ approximation in expectation.*

Proof When a node is selected in line 6 in Algorithm 4, we charge 1, and the charge is evenly divided among all the newly covered nodes. In line 9, another node is chosen with probability $p = \frac{1}{\sqrt{H(d)}}$, where d is the largest degree we currently know. Thus whenever S_i receives charge c at some step, the expected number of nodes in our solution increases by $c(1+p)$. Note that this p changes over time.

The analysis for each S_i is divided into two phases. The first phase ends when one of the nodes in S_i gets chosen. In other words, we come to the second phase when i is within the 1-hop neighborhood of some chosen node. We start considering the first phase. Let X_j be a random variable indicating whether

a node in S_i is chosen at step j ($1 \leq j \leq n$). If we denote the total charge S_i received at this step as p_j, and the corresponding probability of choosing another random neighbor as p'_j, then $E[X_j] = p_j \cdot p'_j$ conditioned on p_j. The first phase will end no later than step j if X_j is the first to be 1 (if a node in S_i is chosen greedily, it will only stop first phase faster). Thus we need to define a stopping time T to bound the total increase in expected number of nodes.

Definition 5 Let T be the random variable denoting the smallest j such that $X_j = 1$ (or n if $X_j = 0$ for all j).

With the stopping time T in hand, we can write out the total increase in expected number of nodes in the first phase as $\mathbb{E}_T\left[\sum_{j=1}^{T} p_j + \sum_{j=1}^{T} p_j \cdot p'_j\right]$, which is bounded by the following lemma,

Lemma 3

$$\mathbb{E}_T[\sum_{j=1}^{T} p_j(1+p'_j)] \leq 1 + \sqrt{H(\Delta)}.$$

Proof We prove it by induction on n. If $n = 1$, the left side is $p_1 + p_1 p'_1 \leq 1 + 1 = 1 + \sqrt{H(1)} < 1 + \sqrt{H(\Delta)}$, which is trivial. Suppose it holds for $n = k - 1$, we prove it holds for $n = k$

$$\begin{aligned}
\mathbb{E}_T\left[\sum_{j=1}^{T} p_j(1+p'_j)\right] =& p_1(1+p'_1) + \Pr[X_1 = 0]\mathbb{E}_T\left[\sum_{j=2}^{T} p_j(1+p'_j) | X_1 = 1\right] \\
=& p_1(1+p'_1) + (1-p_1p'_1)\mathbb{E}_T\left[\sum_{j=2}^{T} p_j(1+p'_j) | X_1 = 1\right] \\
\leq& p_1(1+p'_1) + (1-p_1p'_1)\left(1+\sqrt{H(\Delta)}\right) \\
=& p_1 + p_1p'_1 + 1 + \sqrt{H(\Delta)} - p_1p'_1 - p_1p'_1\sqrt{H(\Delta)} \\
\leq& 1 + p_1 + \sqrt{H(\Delta)} - p_1\frac{1}{\sqrt{H(\Delta)}}\sqrt{H(\Delta)} \\
=& 1 + p_1 + \sqrt{H(\Delta)} - p_1 \\
=& 1 + \sqrt{H(\Delta)}
\end{aligned}$$

The first inequality is applying induction hypothesis on $X_2, \cdots, X_n$, and the second inequality uses the fact that $p'_j = \frac{1}{\sqrt{H(\delta(v))}}$ for some node v, and $\delta(v) \leq \Delta$. So $p'_j \geq \frac{1}{\sqrt{H(\Delta)}}$. □

As for the second phase, whenever S_i receives charge c, on expectation, the total number of nodes will increase by $c(1+p)$, $p = \frac{1}{\sqrt{H(d)}}$, where d is the largest degree we currently know. Since $d_i = \delta(i)$ is already known, we have

$d \geq d_i$. So $p = \frac{1}{\sqrt{H(d)}} \leq \frac{1}{\sqrt{H(d_i)}}$, which implies that the increase in expected number of nodes in S_i is bounded by $\frac{c}{\sqrt{H(d_i)}}$.

The total charge is bounded by $H(d_i)$. This can be done using the standard technique in set cover proof [3]. Thus the total number of nodes chosen in phase 2 is bounded by

$$\begin{aligned} H(d_i)\,(1+p) \leq & H(d_i)\left(1+\frac{1}{\sqrt{H(d_i)}}\right) = H(d_i) + \frac{H(d_i)}{\sqrt{H(d_i)}} = H(d_i) + \sqrt{H(d_i)} \\ \leq & \sqrt{H(\Delta)} + H(\Delta) \end{aligned}$$

Combining the charge from both phases, the expected number of nodes chosen in S_i is bounded by $H(\Delta) + 2\sqrt{H(\Delta)} + 1$. This directly means that the expected size of solution is bounded by $|OPT| \cdot (H(\Delta) + 2\sqrt{H(\Delta)} + 1)$, implying $H(\Delta) + 2\sqrt{H(\Delta)} + 1$ approximation. □

5 Future work

With only local information, we get almost the same approximation ratio as when we have global information, for the connected dominating set problem. Is it also the case for other problems? Or does the lack of information prove to be a huge obstacle for designing algorithms?

Our first algorithm requires information within 2-hops. When only 1-hop local information is available, we cannot get the same result. Compared with previous result [2], the speed and approximation ratio is improved, i.e. $(H(\Delta) + 2\sqrt{H(\Delta)} + 1)$. But a gap still persists. Can we do better? Or is this the price we pay for lack of information?

References

1. Konstantin Avrachenkov, Prithwish Basu, Giovanni Neglia, Bernardete Ribeiro, and Don Towsley. Pay few, influence most: Online myopic network covering. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 813–818. IEEE, 2014.
2. Christian Borgs, Michael Brautbar, Jennifer Chayes, Sanjeev Khanna, and Brendan Lucier. The power of local information in social networks. In *Internet and Network Economics*, pages 406–419. Springer, 2012.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
4. Ding-Zhu Du and Peng-Jun Wan. *Connected dominating set: theory and applications*, volume 77. Springer Science & Business Media, 2012.

5. Devdatt Dubhashi, Alessandro Mei, Alessandro Panconesi, Jaikumar Radhakrishnan, and Aravind Srinivasan. Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 717–724. SIAM, 2003.
6. Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
7. Sudipto Guha and Samir Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.
8. Sudipto Guha and Samir Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Information and computation*, 150(1):57–74, 1999.
9. Samir Khuller, Manish Purohit, and Kanthi K Sarpatwar. Analyzing the optimal neighborhood: algorithms for budgeted and partial connected dominating set problems. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1702–1713. SIAM, 2014.
10. Yuzhen Liu and Weifa Liang. Approximate coverage in wireless sensor networks. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 68–75. IEEE, 2005.
11. Adish Singla, Eric Horvitz, Pushmeet Kohli, Ryen White, and Andreas Krause. Information gathering in networks via active exploration. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 981–988. AAAI Press, 2015.