A Unified Approach for Domination Problems on Different Network Topologies

Thang N. Dinh, Dung T. Nguyen, My T. Thai

Abstract We provide tight hardness results and approxiamation algorithms for many existing domination problems. We start with the *positive influence dominating set* (PIDS) problem, originated from the context of influence propagation in social networks. The PIDS problem seeks for a minimal set of nodes $\mathscr{P}$ such that all other nodes in the network have at least a fraction $\rho > 0$ of their neighbors in $\mathscr{P}$; in the *total* version (T-PIDS), nodes in $\mathscr{P}$ are required to have a fraction ρ of neighbors inside $\mathscr{P}$; and in the *connected* version (C-PIDS) the dominating set have to induce a connected subgraph. Then, we unify a large variations of dominating set problem under a single generalized dominating set problem.
We show a tight hardness results $\frac{1}{2}(1-o(1))\ln n$ inapproximability and a $\ln\Delta + O(1)$ approximation algorithms for our generalized dominating set problem where n is the newtork size and Δ is the maximum degree. The results apply directly to PIDS, k-tuple dominating set, m-connected k-dominating set, Fixed Threshold Dominating Set and many existing domination problems plus all connected or/and total versions of those problems. As most previous hardness results are NP-completeness or APX-hardness, we effectively close many long-standing approximation gaps of domination problems, under the reasonable assumption that NP $\not\subset$ DTIME($n^{O(\log\log n)}$). In networks with degrees bounded by a constant B, we show that all problems cannot be approximated within $\ln B - O(\ln\ln B)$, unless P=NP. In dense networks and scale-free networks such as Internet, WWW, social networks, etc. in which degree sequences follows a power-law distribution, we reveal trivial constant factor approximation algorithms for the class of PIDS-like domination problems. Finally, we prove that optimal solution of any domination problems can be found in linear time for networks with tree topology.

Thang N. Dinh
University of Florida, Gainesville, FL 32611, US; e-mail: `tdinh@cise.ufl.edu`

Dung T. Nguyen
University of Florida, Gainesville, FL 32611, US; e-mail: `dtnguyen@cise.ufl.edu`

My T. Thai
University of Florida, Gainesville, FL 32611, US; e-mail: `mythai@cise.ufl.edu`

1 Introduction

Given a graph $G = (V,E)$, a dominating set (DS) is a subset $\mathscr{D}$ of V such that every vertex not in $\mathscr{D}$ is joined to at least one vertex in D by some edge. Finding minimum dominating set has been extensively studied and applied to many practical applications, for examples, building virtual backbones for routing, power-management, topology control, broadcast scheduling in wireless networks[1, 2, 3, 4, 5, 6, 7]. Recently, variations of dominating set have found intersting applications in context of social networks [8, 9, 10].

1.1 Positive Influence Dominating Set

Regularly, individuals tend to be influenced by the opinions/behaviors of their relatives, friends and colleagues. For examples, children whose parents smoked are twice as likely to begin smoking between 13 and 21 [11], and peer pressure accounts for 65% reasons for binge drinking, a major health issue, by children and adolescents [12]. Moreover, the tendency of an user to adopts a behavior increases together with the number his neighbors follows that behavior.

Exploiting the relationships and influences among individuals in social networks might offer considerable benefit to both the economy and society. As an example, positive impacts of intervention and education programs on a properly selected set of initial individuals can diffuse widely into society via various social contacts: face-to-face, phone calls, email, social networks and so on. How to select a subset of individuals to be included into intervention programs in order to spread the positive effect through the whole targeted group is an important research problem.

Findind a proper subset of most influential individual is formulated into a domination problem [8] in which an individual in the network becomes “influenced” if half of its neighbors are “positive” about adopting a product or behavior. Formally, let $G = (V,E)$ be a graph that represents the social network and a influence factor $0 < \rho < 1$, we wish to find a minimum set of core vertices $\mathscr{P}$, so that every other vertices in G has at least a fraction ρ of their neighbors in $\mathscr{P}$. We denote by $N(v)$ the set of neighbors of a vertex $v \in V$ and $d(v) = |N(v)|$ the degree of v.

POSITIVE INFLUENCE DOMINATING SET (PIDS)

Input: An undirected graph $G = (V,E)$ with influence factor $0 < \rho < 1$.

Problem: Find a subset $\mathscr{P} \subset V$ such that $\forall u \in V \setminus \mathscr{P} : |N(u) \cap \mathscr{P}| \geq \rho d(u)$.

We say nodes in $\mathscr{P}$ dominate or influence their neighbors in $V \setminus \mathscr{P}$. The constant ρ is called the *influence factor*, since it determines for each node the minimum number of neighbors to include in the PIDS. The studied problem in [8, 9, 10] is a special case of the above problem with $\rho = 1/2$.

If we require even nodes in the PIDS $\mathscr{P}$ to be dominated by a fraction ρ of their neighbors, we have the total version of the problem.

TOTAL POSITIVE INFLUENCE DOMINATING SET (T-PIDS)

Input: An undirected graph $G = (V, E)$ with influence factor $0 < \rho < 1$.

Problem: Find a subset $\mathscr{P}_T \subset V$ such that $\forall \mathbf{u} \in \mathbf{V} : |N(u) \cap \mathscr{P}_T| \geq \rho d(u)$.

In the connected domination variation, we wish to find a set of connected vertices that form a positive influence dominating set.

CONNECTED POSITIVE INFLUENCE DOMINATING SET (C-PIDS)

Input: An undirected connected graph $G = (V, E)$ with influence factor $0 < \rho < 1$.

Problem: Find a subset $\mathscr{P}_C \subset V$ such that $\forall u \in V \setminus \mathscr{P}_C : |N(u) \cap \mathscr{P}_C| \geq \rho d(u)$ and $\mathscr{P}_C$ induces a **connected** subgraph of G.

Similarly, *Total Connected Positive Influence Dominating Set* (TC-PIDS) problem asks for an connected and total PIDS $\mathscr{P}_{TC}$ of G.

1.2 Generalized Dominating Set

We generalize the domination requirement $\rho d(u)$ for vertex u with a predefined function $r_u(x)$ of $d(u)$, called the *threshold function*.

GENERALIZED DOMINATING SET (GDS)

Input: An undirected graph $G = (V, E)$ and a family of functions $r_v(x)$ for $v \in V$.

Problem: Find a subset $\mathscr{P} \subset V$ such that $\forall u \in V \setminus \mathscr{P} : |N(v) \cap \mathscr{P}| \geq r_v(d(v))$.

Different family of $\{r_v(x)\}_{v \in V}$ corresponds to different domination problems, for examples, $r_v(x) = 1$ gives the usual dominating set, $r_v(x) = \rho x$ gives the PIDS problem, $r_v(x) = t$, where t is a positive constant, gives the t-TUPLE DOMINATING SET problem[13, 14], $r_v(x) = c_v$, where c_v are positve constants, gives the FIXED THRESHOLD DOMINATING SET (FTDS) problem. Moreover, for scale-free networks that degree sequences follow power-law distribution, we might anticipate the use of threshold functions such as $r_v(x) = \sqrt{x}$ or $r_v(x) = \log x$, .etc.

Notice that using a function $r_v(x)$ instead of a simple constant c_v for vertex v (the case of FTDS) allows the domination requirement of v to be adjusted accordingly when the network evolves or is argumented. It is important to treat GDS as a family of problems but not a single problem. Different families of threshold functions might lead to very different approximation algorithm and inapproximability results. For example, setting $r_v(x) = x$ yields the MINIMUM VERTEX COVER that has simple 2-approximation algorithm and $2 - \varepsilon$ lower-bound under the unique games conjecture [15], while most domination problems achieve only $O(\log n)$ approximation algorithms.

In the same way that T-PIDS, C-PIDS, TC-PIDS are formulated, we also define:

- TOTAL GENERALIZED DOMINATING SET (T-GDS),
- CONNECTED GENERALIZED DOMINATING SET (C-GDS),

- k-CONNECTED GENERALIZED DOMINATING SET (kC-GDS), in which the subgraph induced by the dominating set is k-vertex connected i.e. there are at least k vertex-disjoint paths betwen any vertices in the dominating set, where k is an integral constant. C-GDS is a special case of kC-GDS with $k = 1$.
- TOTAL CONNECTED GENERALIZED DOMINATING SET (TC-GDS),
- k-CONNECTED TOTAL GENERALIZED DOMINATING SET (kCT-GDS),

Threshold function. Not all threshold functions yield meaningful domination problems. For example, if $r_v(x) > x$, then there is no way to satisfy the requirement $|N(v) \cap \mathscr{P}| > r_v(d(v)) > d(v)$. Hence, we restrict our attention to the following class of threshold function that is general enough to covered existing and possibly incoming domination problems.

Definition 1. A function $r_v(x) : \mathbb{N} \to \mathbb{R}^+$ is a *dominating function* if and only if $r_v(x)$ is a *monotone increasing* function that satisfies the following conditions.

1. $0 < r_v(x) < x \quad \forall x > 0$,
2. $0 \leq r_v(x) - r_v(x-1) \leq 1 \quad \forall x > 0$,
3. $\lim_{x\to\infty} \dfrac{r_v(x)}{x} < 1$.

The first condition makes the problem "non-trivial" to solve (to be precise it makes the problem hard to aproximate within $O(\log n)$). The second and third conditions guarantee the growing rate of the function to be linear or sublinear.

1.2.1 Related Work

Finding minimum dominating set is hard to approximate within $(1 - o(1)) \ln n$ by a reduction to Set cover [16] and is approximable within $\ln \frac{|\mathscr{S}|}{\mathrm{OPT}} + O(\ln \ln \frac{|\mathscr{S}|}{\mathrm{OPT}})$ [17]. On special graphs such as planar graphs, disk graphs, there exist PTASs [18, 19, 20, 21, 22, 23]. In the context of wireless sensor networks (WSNs), distributed $(1+\varepsilon)$ approximation algorithm on Unit Disk Graphs (UDGs) with $O(\log^* |G|)$ rounds are known [24]. When vertices are associated with weights or costs, $5+\varepsilon$ approximation algorithm is known for UDG [25].

Henning et al. [26] survey the total dominating set problem and provideNP-hard proofs on some special graphs.

The connected dominating set problem is studied in Guha and Khuller [27] in which a $H(\Delta)+2$ approximation algorithm for finding minimum CDS is introduced. In UDGs, constant factor approximation algorithms for weighted connected DS are possible. The currently best ratio is $9+\varepsilon$ presented in [25].

The t-tuple DS is studied in [28, 13, 14] in which the approximation lower bound of $(1-\varepsilon)\ln n$ together with $\ln(\Delta + 1) + 1$ approximation algorithm were presented. Algorithms for minimum k-connected t-tuple dominating set problem [29] $6 + \ln 5/2(k-1) + 5/k)$ approximation algorithm on UDG. No hardness results other than NP-completeness for the problem were presented in literature.

Unless NP $\subseteq$ DTIME$\left(n^{O(\log\log n)}\right)$, Feige [16] proved that Set cover cannot be approximated within factor of $(1-o(1))\cdot\ln n$, while under the typical assumption that P $\neq$ NP, the best known inapproximability result is $c\cdot\ln n$ [30] for some constant $c>0$.

Domingos and Richardson [31] were the first to study the propagation of influence and the problem of identification of the most influential users in networks. Kempe et al. [32, 33] formulated the influence maximization problem as an optimization problem. Leskovec et al. [34] study the influence propagation in a different perspective in which they aim to find a set of nodes in networks to detect the spread of virus as soon as possible.

Influence propagation with a limited number of hops as well as a special case of T-PIDS, when $\rho = 1/2$ were first considered in Wang et al. [8] in which they iteratively add (normal) dominating sets until forming a T-PIDS. Feng et al. [35] showed NP-completeness for the PIDS problem, when $\rho = 1/2$. The APX-hardness and an $O(\log n)$ approximation algorithm for T-PIDS problem was introduced in [9]. Under the condition that the geometric representation of a Unit Disk Graph is given and the maximum degree is bounded by a constant, Zhang et al. [36] devised a Polynomial Time Approximation Scheme (PTAS) for the t-latency bounded information propagation.

1.2.2 Our Results

Problem	Original	T-(*)	kC-(*)	kCT-(*)
t-tuple DS	$(1-\varepsilon)\ln n$[13]	$(1-\varepsilon)\ln n$	$(1-\varepsilon)\ln n$, $k=1$ $k=1$ [13], $\forall k$[$\star$]	$(1-\varepsilon)\ln n$, $k=1$ $k=1$ [13], $\forall k$[$\star$]
Positive Influence DS	APX-hard, $\rho=\frac{1}{2}$ [10] $(1-\varepsilon)\ln\Delta[\star]$ $(\frac{1}{2}-\varepsilon)\ln n[\star]$	- $(1-\varepsilon)\ln\Delta[\star]$ $(\frac{1}{2}-\varepsilon)\ln n[\star]$	APX-hard, $k=1$[9] $(1-\varepsilon)\ln\Delta[\star]$ $(\frac{1}{2}-\varepsilon)\ln n[\star]$	- $(1-\varepsilon)\ln\Delta[\star]$ $(\frac{1}{2}-\varepsilon)\ln n[\star]$
Fixed Threshold DS	$(1-\varepsilon)\ln n$	$(1-\varepsilon)\ln n$	$(1-\varepsilon)\ln n[\star]$	$(1-\varepsilon)\ln n[\star]$

Table 1: Hardness of Approximation for Domination Problems in graphs. Symbol $[\star]$ means results are derived from this chapter. The hardness $(1-\varepsilon)\ln\Delta$ is proved under the typical assumption P $\neq$ NP, while the hardness $(\frac{1}{2}-\varepsilon)\ln n$ is proved with the assumption NP $\not\subset$ DTIME$(n^{O(\log\log n)})$.

Following results are presented in the chapter.

- We prove that domination problems belong to the families GDS, T-GDS, C-GDS, and TC-GDS can be approximated within $\ln\Delta + O(1)$ but cannot be approximated within $(\frac{1}{2}-o(1))\ln n$, unless NP $\subset$ DTIME$(n^{O(\log\log n)})$.

Problem	Original	T-(*)	kC-(*)	kCT-(*)
Dominating Set	$\ln B - O(\ln\ln B)$[37]	$\ln B - O(\ln\ln B)$[37]	$\ln B - O(\ln\ln B)$ $k = 1$ [37], $\forall k[\star]$	$\ln B - O(\ln\ln B)$ $k = 1$ [37], $\forall k[\star]$
t-tuple DS	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$
Positive Influence DS	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$
Fixed Threshold DS	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$	$\ln B - O(\ln\ln B)[\star]$

Table 2: Hardness of Approximation for Domination Problems in B-bounded graphs. Symbol $[\star]$ means results are derived from this chapter. The hardness are proved with the assumption P $\neq$ NP.

- On B-bounded degree graphs, none of the problems can be approximated within $\ln B - O(\ln\ln B)$, under the standard assumption $P \neq NP$. As a consequence, the considered problems are not approximated within $\ln\Delta - O(\ln\ln\Delta)$, unless P=NP.
- If the network is scale-free i.e. the degree sequence follows a power-law distribution or the network is dense, we analyze the degree-based greedy selection algorithm to show that it obtains constant approximation algorithm for "PIDS-like" domination problems.
- In networks with tree topologies, it is possible to find the optimal solution in linear-time for all considered domination problems.

We also summarize implied results in this chapter in Tables 1 and 2.

2 Hardness of Approximation

In this section, we present inapproxiability results for domination problems in families GDS, T-GDS, kC-GDS, TC-GDS. To make the chapter easier to follow, we first present the proof for B-bounded graphs in subsection 2.2, then extend the gadget in the proof to obtain the hardness results in general graphs in subsection 2.3.

Our proof requires fine-tuning settings in Feige's reduction for Set Cover [16], an example of refined elegance. Although, we can use hardness results of Set Cover in a black-box fashion (or equilvalently dominating set problem), it leads to weaker inapproximability $O(\log B)$ and $O(\log n)$ for B-bounded graph and general graph, respectively, that is a constant time worse than the tight hardness results.

The challenge lies on bounding the size of added vertices in our reductions with the size of the optimal set cover. Two important quantities that are not mentioned in the Feige's reduction are the maximum capacity of a set and the maximum frequency of an point. We brief the Feige's proof for Set cover in subsection 2.1 and derive bounds on latent parameters to help in the analysis of our reductions.

2.1 Feige's Reduction for Set Cover

Feige presented a reduction from a k-prover proof system for a MAX 3SAT-5 instance ϕ that is a *conjunctive normal form* formula consists of n varilables and $\frac{5n}{3}$ clauses of exactly 3 literals. The verifier interacts with k provers, and ask provers different questions based on a random string r; each question involves $l/2$ clauses and $l/2$ variables. If the formula ϕ is satisfiable, then the provers have a strategy that cause the verifer accepts for all random strings. If only a $(1-\varepsilon)$ fraction of the clauses in ϕ are simultaneously satisfiable, then for all strategies of the provers, the verifer weakly accept with a probability at most $k^2 \cdot 2^{-cl}$, where c is a constant that depends only on ε.

The core of the Set cover gadget is a partition system $B(m,L,k,d)$, where B is a ground set of m points. The partition system is a collection of $L = 2^l$ partitions $P_1,\ldots,P_L$ of B, each partition P_i has exactly k disjoint subsets $p_{i,1},\ldots,p_{i,k}$. Any cover of m points in B requires at least $d = (1-\frac{2}{2})k\ln m$ subsets. The condition to make constructing such a system possible is that $k < \frac{\ln m}{3\ln\ln m}$.

Let $R = (5n)^l$ denote the number of possible random strings for the verifier. We make R copies of parition system B. Let B_r denote the copy of the partition associated with the random string r and $p^r_{i,j}$ the copy of set $p_{i,j}$ in B_r.

We now ready to describe the instance of Set Cover in the Feige's reduction. The univeral set $\mathscr{U} = \bigcup\limits_{r\in R} B_r$ contains mR points; and the set system is $\mathscr{S} = \left\{S_{q,a,i}\right\}_{q,a}$, where i can be deduced from syntax of (q,a). Each set $S_{q,a,i}$ corresponds to a question-answer pair (q,a) of the ith prover and $S_{q,a,i} = \bigcup\limits_{(q,i)\in r} p^r_{a_r,i}$ where $(q,i)\in r$ means on random string r, the ith prover receives question q, and a_r is the assignment of variables extracted from a.

As long as $k^2 2^{-cl} < \frac{8}{k^3\ln^2 m}$, we obtain the hardness result $(1-\frac{4}{k})\ln m$ i.e. if formula ϕ is satisfiable, then mR points in $\mathscr{U}$ can be covered by kQ subsets, and if only $(1-\varepsilon)$ fraction of the clauses are simultaneously satisfiable, the minimum set cover has size at least $(1-\frac{4}{k})\ln m\ kQ$. Here, Q is the set of all $n^l\,(5/3)^{l/2}$ possible questions. The condition can be satisfied with $l > \frac{1}{c}(5\log k + 2\log\ln m)$.

We now present important quantities that appear later in our proofs.

- $|\mathscr{S}| \le Q2^{2l}$: For each question $q\in Q$, there are at most 2^{2l} answers of $2l$ bit length.
- $\Delta_{\mathscr{S}} = \max\limits_{S\in\mathscr{S}} |S| \le m3^{l/2}$: For each i and $q\in Q$ there are at most $3^{l/2}$ random strings r such that the verifier makes query q to the ith prover and $|p^r_{a_r,i}| \le m$.
- $\mathrm{F}_{\mathscr{U}} \le k2^l$: Where $\mathrm{F}_{\mathscr{U}}$ is the maximum frequency of a point in $\mathscr{U}$. Because, for a pair (q,i), each partition $p^r_{a_r,i}$ is included at most 2^l times, plus each point in B_r appears in exactly k paritions.

2.2 *Tight Hardness Results on Bounded-Degree Graphs*

The hardness results on bounded-degree graphs are divided two parts. In the first part, hardness resutls for PIDS and its variations are established. In the second part, the hardness of domination problems in the families GDS, T-GDS, *k*C-GDS, *d*BD-GDS are proved by subtle modifcations on reductions in the first part.

2.2.1 Positive Influence Dominating Set

Theorem 1. *Neither PIDS, T-PIDS, kC-PIDS can be approximated within* $\ln B - O(\ln\ln B)$ *in B-bounded graphs, unless P=NP.*

We use a reduction from an instance of the *Bounded Set Cover* problem (SC_B) to an instance of PIDS problem whose degrees are also bounded by $B' = B$ poly log B.

BOUNDED SET COVER (SC_B)

Input: A set system $(\mathscr{U}, \mathscr{S})$, where $\mathscr{U} = \{e_1, e_2, \ldots, e_n\}$ is a universe and $\mathscr{S}$ is a collection of subsets of $\mathscr{U}$. Each subset in $\mathscr{S}$ has at most B elements and each point belongs to at most B subsets, for a predefined constant $B > 0$.

Problem: Find a cover that is a subfamily $\mathscr{C} \subseteq \mathscr{S}$ of sets whose union is $\mathscr{U}$ with the minimum number of subsets.

For a sufficient large constant $B_0 > 0$, set cover problem where each set has at most $B > B_0$ elements is hard to approximate to within a factor of $\ln B - O(\ln\ln B)$, unless $P = NP$ [38].

The proof [38] maps an instance of $GAP-SAT_{1,\gamma}$ to an instance $\mathscr{F} = (\mathscr{U}, \mathscr{S})$ of set cover with $\Delta_S \le B$. Parameters l, m in Feige's construction [16] are fixed to $\theta(\ln\ln B)$ and $\frac{B}{\text{poly}\log(B)}$, respectively. Since the parameters can be found in constant time using brute-force search, the assumption for the hardness is P $\neq$ NP instead of NP $\not\subset$ DTIME($n^{O(\log\log n)}$). The produced instance has the following properties that will be used later in our proofs.

- $|\mathscr{U}| = mn^l$ poly log$B, |\mathscr{S}| = n^l$ poly log B
- $\Delta_{\mathscr{S}} \le B, F_{\mathscr{U}} \le$ poly log B for sufficient large B.

A sufficient large constant B_0 gives us $f \le$ poly $\log(B) \le B$ for all $B \ge B_0$. $\square$

SC_B-PIDS reduction. For each instance $\mathscr{F} = (\mathscr{U}, \mathscr{S})$ of SC_B, we construct a graph $\mathscr{H} = (V, E)$ as follows (Fig. 1):

- Construct a bipartite graph with the vertex set $\mathscr{U} \cup \mathscr{S}$ and edges between S and all elements $x \in S$, for each $S \in \mathscr{S}$.
- Add a set D consisting of t vertices and a set D' with same number of vertices, say $D = \{x_1, x_2, \ldots, x_t\}$ and $D' = \{x'_1, x'_2, \ldots, x'_t\}$. The value of t will be determined later.
- Connect x_i to $x'_i, \forall i = 1 \ldots t$ to force the selection of x_i in the optimal PIDS.

- Connect each vertex $e_j \in \mathscr{U}$ to $\lceil \frac{\rho}{1-\rho} f(e_j) \rceil - 1$ and each vertex $S_k \in \mathscr{S}$ to $\lceil \frac{\rho}{1-\rho} |S_k| \rceil$ vertices in D, where $f(e_j)$ is the frequency of point e_j. During the connection, we balance the degrees of vertices in D.

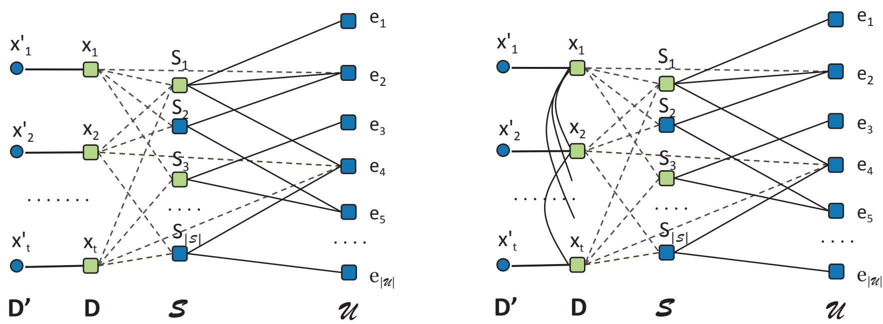

Fig. 1: Reduction from SC_B to PIDS (left) and T-PIDS (right)

Lemma 1. *The size difference between the optimal PIDS of $\mathscr{H}$ and the optimal* SC_B *of $\mathscr{F}$ is exactly the cardinality of D, i.e.,* $\text{OPT}_{\text{PIDS}}(\mathscr{H}) = \text{OPT}_{SC}(\mathscr{F}) + t$.

Proof. Let $\mathscr{P}$ be an optimal PIDS of $\mathscr{H}$. Since either x_i or x'_i must be selected into $\mathscr{P}$, and we can always replace $x'_i \in \mathscr{P}$ with x_i inside $\mathscr{P}$. Thus, it is safe to assume that $D' \cap \mathscr{P} = \emptyset$ and $D \subset \mathscr{P}$.

By the construction, each vertex $S_k \in \mathscr{S}$ has enough required neighbors in $\mathscr{P}$, while each vertex $e_i \in \mathscr{U}$ needs at least one more neighbor in $\mathscr{P}$ or it has to be selected. Since all vertices in $\mathscr{U}$ must be adjacent to at least one vertex in $\mathscr{S}$, we can always replace each vertex $e_i \in \mathscr{P}$ with one of its neighbor in $\mathscr{S}$ without increasing the size of $\mathscr{P}$. We therefore can assume that the optimal solution will contains vertices in $\mathscr{S}$ but not in $\mathscr{U}$.

Hence, $\mathscr{P} \setminus D$ must induce a cover for $\mathscr{F} = (\mathscr{U}, \mathscr{S})$. In other words, we have $\text{OPT}_{\text{PIDS}}(\mathscr{H}) \geq \text{OPT}_{SC}(\mathscr{F}) + t$.

Besides, given a cover $\mathscr{C} \subseteq \mathscr{S}$ for $(\mathscr{U}, \mathscr{S})$, it is easy to check that $\mathscr{C} \cup D$ gives a PIDS for $\mathscr{H}$. Thus, $\text{OPT}_{SC}(\mathscr{F}) + t \geq \text{OPT}_{\text{PIDS}}(\mathscr{H})$ that completes the proof. □

The requirements to transfer hardness results of set cover to PIDS problem is to keep the degree of vertices in $\mathscr{H}$ bounded by B' and keep t sufficiently small in comparison with the optimal solution of the SC_B instance in order to derive the ratio.

Lemma 2. *There exists a construction of $\mathscr{H}$ with $t \leq \frac{\text{OPT}_{\text{SC}}}{\ln^2 B}$ and $B' = \Delta(\mathscr{H}) = O(B \text{ poly} \log B)$.*

Proof. We first compute $\text{vol}(D)$, the total degree of vertices in D. For two sets of vertices A and B, we define $\phi(A, B)$ the set of edges crossing between them.

$$\begin{aligned}\mathrm{vol}(D) &= |\phi(D,D')| + |\phi(D,\mathscr{U})| + |\phi(D,\mathscr{S})| \\ &= |D| + \sum_{S_k\in\mathscr{S}} \lceil \frac{\rho}{1-\rho}|S_k| \rceil + \sum_{e_j\in\mathscr{U}} \lceil \frac{\rho}{1-\rho} f(e_j) - 1 \rceil \\ &\le \frac{2\rho}{1-\rho}|\mathscr{S}|B + |\mathscr{S}| + t = \left(\frac{2\rho}{1-\rho}B+1\right)|\mathscr{S}| + t \qquad (1)\end{aligned}$$

We have used the facts that $\sum_{S_k\in\mathscr{S}} |S_k| = \sum_{e_j\in\mathscr{U}} f(e_j)$ and $|S_k| \le B,\ \forall S_k \in \mathscr{S}$.

Select $t = \frac{|\mathscr{U}|}{B\ln^2 B}$. Since each set in $\mathscr{S}$ can cover at most B elements, it follows that $\mathrm{OPT}_{SC} \ge \frac{|\mathscr{U}|}{B}$, hence, $\frac{\mathrm{OPT}_{SC}}{\ln^2 B} \ge t$.

To have a valid construction of $\mathscr{H}$, it is sufficient that $t.B' \ge \mathrm{vol}(D)$. Thus, we select B' satisfying

$$\begin{aligned}B' &\ge \frac{1}{t}\left(\left(\frac{2\rho}{1-\rho}B+1\right)|\mathscr{S}| + t\right) \\ &\approx \left(\frac{2\rho}{1-\rho}B+1\right)\frac{B\ln^2 B\, n^l \text{ poly log } B}{mn^l \text{ poly log } B} \approx B \text{ poly log } B \qquad (2)\end{aligned}$$

Hence, setting $B' = B$ poly log B gives us the desired construction of $\mathscr{H}$. □

Theorem 2. *There exist constants B_1, c_1 such that for every $B' \ge B_1$ it is NP-hard to approximate the PIDS problem in graphs with degrees bounded by B' within a factor of* $\ln B' - c_1 \ln\ln B'$.

Proof. We prove by contradiction. Assume we have an algorithm that find a PIDS of size at most $\ln B' - c_1 \ln\ln B'$ the optimal size in graph with degrees bounded by B'. We then show how to approximate the SC_B problem with ratio $\ln B - c_0 \ln\ln B$ in polynomial time. Selecting sufficient large B_1 is not difficult and shall be ignored to make the proof simpler.

Let $\mathscr{F} = (\mathscr{U}, \mathscr{S})$ be an instance of SC_B. Construct an instance $\mathscr{H}$ of PIDS problem using the reduction SC_B-PIDS. From (2), there exists constant $\beta > 0$ so that $B' \le B\ln^\beta B$. Using the approximation for PIDS, we obtain a solution of size at most $(\ln B' - c_1 \ln\ln B')\mathrm{OPT}_{PIDS}$. We can then convert that to a solution of SC_B by excluding vertices in D (see Lemma 1) and obtain a cover of size at most

$$\begin{aligned}&(\ln B' - c_1 \ln\ln B')(\mathrm{OPT}_{SC} + t) - t \\ \le&(\ln B' - c_1 \ln\ln B')\mathrm{OPT}_{SC} + (\ln B' - c_1 \ln\ln B')\frac{\mathrm{OPT}_{SC}}{\ln^2 B} \\ \le&\left(\ln B + \beta \ln\ln B - c_1 \ln(\ln B + \theta(\ln\ln B)) + O(\frac{1}{\ln B})\right)\mathrm{OPT}_{SC}\end{aligned}$$

Select $c_1 = c_0 + \beta + 1$. The solution for SC_B problem is then smaller than $\ln B - c_0 \ln\ln B$ times $\mathrm{OPT}_{\mathrm{SC}}$ which implies P=NP. □

Theorem 3. *It is NP-hard to approximate T-PIDS, and kC-PIDS problems in graphs of bounded degree $B' > B_2$ within a factor of $\ln B' - c_2 \ln\ln B'$ for some constants $B_2, c_2 > 0$.*

Proof. We adjust the reduction SC_B-PIDS to achieve the same hardness result. The following adjustments are made to guarante *total* and *k-connected* properties.

- On top of the subgraph induced by D, construct $(2+O(1))k$-regular Ramanujan graphs so that the subgraph has vertex expansion at least k [39]. At the same time, connect $\mathscr{S}$ to D so that each vertex in $\mathscr{S}$ has at least k neighbors in D. It is easy to check that the subgraph induced by the solution to the dominating set problem is k-vertex connected.
- We connect a vertex $x_i \in D$ with $\lceil \frac{\rho}{1-\rho} d(x_i) \rceil$ other nodes in D, balancing nodes' degrees in D, so that D can dominate themselves. Thus, we roughly multiple the degree of each node in D by a constant. Since $t = \frac{|\mathscr{U}|}{B \ln^2 B} \gg \lceil \frac{\rho}{1-\rho} B' \rceil$, we always have enough vertices in D to connect x_i to.

Fortunately, we can make such adjustments with subtly increasing in the degrees of nodes in D and the rest of the proof goes through straightforwardly. □

2.2.2 General Cases

Theorem 4. *Domination problems in famlies GDS, T-GDS, and kC-GDS with dominating functions $r_v(x)$ satisfying conditions in Def. 1 cannot be approximated within $\ln B - O(\ln\ln B)$ in B-bounded graphs, unless P = NP.*

Proof. Given B-bounded graph $G = (V,E)$ and functions $r_v(x)$ satisfying conditions of a dominating function (see Definition 1). Let $\rho^* = \max_{v \in V} \lim_{x \to \infty} \frac{r_v(x)}{x}$. By the third condition, we have $\rho^* < 1$ and $\rho = \frac{1}{2}(\rho^* + 1) < 1$. Since $\rho > \rho^*$, there exists an absolute constant $x_0 > 0$ such that $r_v(x) < \rho\, x \quad \forall x > x_0$ and $\forall v \in V$. We will assume through the proof that B is sufficiently large so that $l = \theta(\log\log B) > x_0$.

The main idea in the reduction from dominating set with domination requirement $r_v(d(v))$ is to add connections from v to some $a(v)$ more vertices that are guaranteed to be in the optimal solution so that the remain domination requirement on v becomes one for vertices in $\mathscr{U}$ and zero for vertices in $\mathscr{S}$. That is we need to find $x_v \in \mathbb{N}$ so that

$$c_v(x_v) = r_v(d(v) + x_v) - x_v \in \begin{cases} (0,1] & \text{if } v \in \mathscr{U} \\ (-1,0] & \text{if } v \in \mathscr{S} \end{cases} \tag{3}$$

Claim. For $\mu_v = \max\{\lceil \frac{\rho}{1-\rho} d(v) \rceil, x_0\}$, $c_v(\mu_v) \le 0$.

Proof. $c_v(\mu_v) = \rho\,(d(v) + \mu_v) - x_v = \rho(d(v) - \frac{1-\rho}{\rho}\mu_v) < 0$ □

Since

- $c_v(0) = r_v(d(v)) > 0$,
- $\Delta c_v = c_v(x+1) - c_v(x) \in (-1, 0]$,
- $c_v(\mu_v) < 0$.

There exists some $0 \le x_v \le \mu_v$ such that $c_v(x_v)$ satisfies Eq. 3. Notice that x_0 is substantially smaller than B. Hence, the same construction in the case of *PIDS* with influence factor ρ except in the places of vertices in $v \in (\mathscr{S} \cup \mathscr{U})$ where we connect with x_v vertices in D will work for our general cases as well. □

As an corollary of the hardness result for the bounded degree case, we also have

Theorem 5. *Unless P=NP, domination problems in families GDS, T-GDS, and kC-GDS cannot be approximated within a factor of* $\ln\Delta - O(\ln\ln\Delta)$*, where* Δ *is the maximum degree.*

2.3 Hardness Results on General Networks

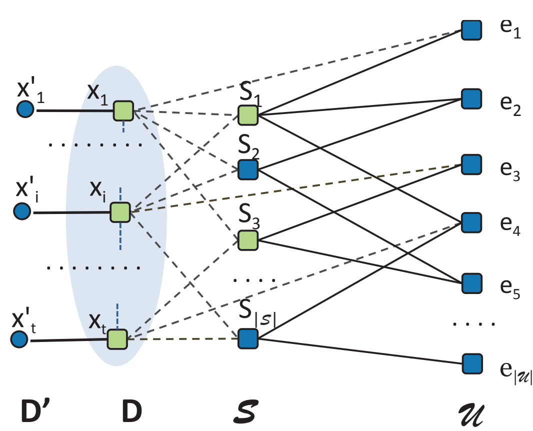

Fig. 2: Reduction from SC_B to GDS, T-GDS, kC-GDS

Theorem 6. *Domination problems in famlies GDS, T-GDS, and kC-GDS with dominating functions* $r_v(x)$ *satisfying conditions in Def. 1 cannot be approximated within* $\frac{1}{2}(1-o(1))\ln n$ *where n is the number of vertices, unless* $\mathrm{NP} \subset \mathrm{DTIME}\left(n^{O(\log\log n)}\right)$.

Proof. We use the same gadget in Fig. 2 to prove for problems in three families . We begin with the gadget in the proof of Theorem 4. Since, we no longer need to keep degree of vertices in the gadget bounded, we form a clique with vertices in D.

Let ρ, x_0, μ_v be the same parameters as in the the proof of Theorem 4. The sufficient conditions to make the construction feasible are

- (GDS): To be able to connect each vertex $v \in (\mathscr{S} \cup \mathscr{U})$ to μ_v vertices in D.

$$|D| = O(\max_{v \in (\mathscr{S} \cup \mathscr{U}) \mu_v} \theta(\frac{\rho}{1-\rho} \Delta_{\mathscr{S}})) = O(\Delta_{\mathscr{S}}) = O(m 3^{l/2})$$

- (kC-GDS): To add at least k edges from each vertex in $\mathscr{S}$ to vertices in D.

$$|D| = O(k|\mathscr{S}|) = O(|\mathscr{S}|) = O(n^l (5/3)^{l/2} 2^{2l}) = O(n^l 2^{\theta(l)})$$

- (T-GDS): Vertices in D have to dominate themselves. Since $\mu_v = \max\{\lceil \frac{\rho}{1-\rho} d(v) \rceil, x_0\}$, this can be satisified when

$$|D| - 1 = O(\frac{\rho}{1-\rho} \frac{1}{|D|} \sum_{v \in (\mathscr{S} \cup \mathscr{U})} \mu_v).$$

  Or equivalently

$$\begin{aligned} |D|^2 &= O(\sum_{v \in (\mathscr{S} \cup \mathscr{U})} d(v) + x_0(|\mathscr{S}| + |\mathscr{U}|) \\ &= O(2 \sum_{v \in \mathscr{U}} d(v) + |\mathscr{S}| + |\mathscr{U}|) = O(mRk2^l) \end{aligned}$$

To summarize, the sufficient condition so that the dominating set of the graph is k-connected and total at the same time is

$$|D| = O(m 2^{\theta(l)} + n^l 2^{\theta(l)} + (mRk2^l)^{1/2}). \tag{4}$$

Notice that, from the proof of Theorem 2, the hardness ratio is in the form $\frac{(1-\frac{4}{k})kQ \ln m + |D|}{kQ + |D|}$. In the Feige's reduction, $|D| = O(\Delta_{\mathscr{S}}) = O((5n)^{\frac{2l}{\varepsilon}} 2^{\theta(l)})$ that yields the $(1-\varepsilon)\ln n$ hardness ratio for Set cover but makes the hardness ratio of the domination problem get arbitrary close to 1. Fortunately, we will be able to reduce the maximum degree by setting $m = (5n)^{cl}$ with a small constant $c > 0$. The consequece is that $\ln m$ is no longer $\ln N + O(1)$, thus, the inapproximability ratio is reduced.

The optimal setting to get the best inapproximability ratio is to set $m = (5n)^{l(1-\varepsilon)}$ for some $\varepsilon > 0$. Then, $N = mR = (5n)^{l(2-\varepsilon)}$, or $m = N^{\frac{1-\varepsilon}{2-\varepsilon}}$. Hence, from (4), it is sufficient that

$$|D| = n^l \frac{2^{\theta(l)}}{n^{l\frac{\varepsilon}{2}}} = o(Q)$$

for sufficiently large l and n. Hence, the hardness ratio will be

$$\frac{(1-\frac{4}{k})kQ \ln m + o(Q)}{kQ + o(Q)} > (1 - \frac{5}{k}) \ln m$$

The number of vertices in the graph is

$$n_{\mathscr{H}} = 2|D| + |\mathscr{S}| + |\mathscr{U}| < \theta(m3^{l/2}) + n^l 2^{2l}\left(\frac{5}{3}\right)^{l/2} + (5n)^{2l-\varepsilon} < 2|\mathscr{U}| = 2N$$

Thus, the hardness ratio is at least

$$(1-\frac{5}{k})\ln\left(\frac{n_{\mathscr{H}}}{2}\right)^{1/2-\frac{\varepsilon}{4-2\varepsilon}} > (1-\frac{5}{k})\frac{1}{2}\left(1-\frac{\varepsilon}{2-\varepsilon}\right)\ln n_{\mathscr{H}} - \theta(1) > \frac{1}{2}(1-\varepsilon)\ln n_{\mathscr{H}}$$

for sufficiently large k and sufficiently small ε. □

Theorem 7. *Domination problems in famlies GDS, T-GDS, and kC-GDS with dominating functions $r_v(x) = O(\log x)$ cannot be approximated within $(1-o(1))\ln n$ where n is the number of vertices, unless* NP $\subset$ DTIME $\left(n^{O(\log\log n)}\right)$.

Proof. When $r_v(x) = O(\log x)$, the size of $|D|$ is only $O(\log \Delta_{\mathscr{S}})$. The original settings in the Feige's reduction will gives the ratio $(1-\varepsilon)\ln n_{\mathscr{H}}$.

3 Approximation Algorithm

3.1 Approximation Algorithm on General Topologies

Theorem 8. *Given a graph $G = (V,E)$, there exist $O((|V|+|E|)\log\log|V|)$ algorithms that approximate GDS within $H(2\Delta)$ and T-GDS within $H(\Delta)$.*

Proof. We begin with the definition of the Constrained Multiset Multicover problem (CMM).

CONSTRAINED MULTISET MULTICOVER (CMM).

Input: A set cover instance $(\mathscr{U},\mathscr{S})$. Each point e has an integer requirement r_e and occurs in a set S with arbitrary multiplicity, denoted $m(S,e)$. Moreover, we associate a cost, c_S, with each set $S \in \mathscr{S}$.

Problem: the minimum cost subcollection which fulfils all elements' cover requirements provided each multiset is picked at most once.

Lemma 3. *[40] There is a natural greedy algorithm that finds a constrained multiset multicover within an H_k factor of the optimal solution, where $k = \max_S \sum_e m(S,e)$.*

The GDS problem on the graph $G = (V,E)$ can be reduced to the following instance of CMM

- $\mathscr{U} = \{e_u : u \in V\}$
- The cover requirement of e_u is set to $r_u = \lceil r_u(d(u)) \rceil$
- $\mathscr{S} = \{S_v : v \in V\}$, where S_v contains $\{e_u : u \in N(v)\}$ plus r_v copies of e_v. That is $m(S_v, e_u) = 1, \forall u \in N(v)$ and $m(S_v, e_v) = \lceil r_v(d(v)) \rceil$.

It follows that the GDS problem can be approximated within

$$H\left(\max_{v\in V} d(v) + r_v(d(v))\right) \leq H(2\Delta) = H(\Delta) + O(1).$$

In case of T-GDS, the only difference in the reduction is that each multiset S_v contains all the neighbors of v, but not any copies of v. The approximation ratio is, hence, $H(\Delta)$.

Implementation Issue: Straightforward implementations might incur high time complexity. In each step, the greedy algorithms select the node with the highest coverage which is bounded by $O(n)$. Hence, using van Emde Boas priority queue[41] to maintain and update nodes' coverages, the total time complexity can be brought down to $O\left((|V| + |E|) \log\log |V|\right)$. Details are presented in the Algorithm 1. $\square$

Corollary 1. *There are polynomial time algorithms that approximate PIDS and T-PIDS problems within ratio* $\ln \Delta + \frac{4}{3}$ *and* $\ln \Delta + 1$*, respectively.*

Proof. Apply Theorem 8 with $r_v(x) = \rho x$ and use the approximation $H(n) \approx \ln n + 0.58$, we can rewrite the approximation ratios for PIDS and T-PIDS as $\left(\ln \Delta + \frac{4}{3}\right)$ and $(\ln \Delta + 1)$.

3.1.1 Extending to d-hop PIDS - VirAds Algorithm

We present an effective implementation for GDS problems called VirAds that also handles the multiple hops PIDS problem.

Cascading in Networks - Referral Model.

We are given a *social network* in the form of an undirected graph $G = (V, E)$ where the vertices V represents users in the network and edges E represents social links between users.

Activations happen in rounds. At a particular round, each vertex is either active or inactive and each node's tendency to become active increases when more of its neighbors become active. Some inactive vertex u will eventually have enough active neighbors to become active, and in turn u's activation will trigger further vertex's activations. Once a vertex becomes active, it will never reverse back to the inactive state.

In the referral model, an inactive vertex v becomes active if number of its active neighbors reaches or exceeds $t(v) = \rho\ degree(v)$, where ρ is the constant in the PIDS problem. It follows the adopting behaviour of a user when a large number of his friends introduce, refer an idea or a product to him. In case $\rho = 1/2$ the model is also known as *majority* that has many application in distributed computing, voting system[42], etc.

We consider the case when all nodes are activated within at most d rounds. The constant d can be customized to adjust the trade off between the size of initial users and how soon and 'fresh' the content reaches users in the network. Study on real network [43] reveals that influence may take many months propagating through the network. By the time reaching the users the content, information might become obsolete or expired. Hence, only users who adopt the product in a given time frame will count. For example it is critical for an political campaign that it influences many people before the election day.

We select vertices one by one favorings ones that can activate many neighbors and ones that requires extra active neighbors to be activated. Since the number of active (inactive) neighbors varies when more and more vertices added, vertices with high degree are not necessary preferred. Our algorithm outperforms Max Degree heuristics that always select the highest degree vertex left in experiments with networks of different scales.

We emphasize that to be applicable for very large OSNs of millions vertices, the key factor is the scalability of the algorithm. Naive implementations of our algorithms will be certainly sluggish and intolerable. After adding a new vertex, the bottleneck arises in browsing for new activated nodes in d rounds as it may cost $O(m+n)$ to do so. Furthermore, P can contain as many as $O(n)$ vertices bringing the algorithm complexity to $O((m+n)\cdot n)$.

We combine proper data structures to devise VirAds as presented in Algorithm 1 with efficient running time $O((m+n)(d+\log\log n))$. Since social networks experience 'small-world phenomenon' , number of propagation rounds d is often small. In our experiments, there are no more vertices activated after 20 rounds when influence factor $\rho > 0.1$. In addition, the factor $\log\log n$ is negligible even for billions n.

For every vertex v, we maintain during the algorithm

- $\mathbf{r_v}$: the round in which v is activated
- $\mathbf{iN_v}$: The number of inactive neighbors of v
- $\mathbf{aN_v}$: The number of extra active neighbors v needs in order to activate v
- $\mathbf{rN_v}[\mathbf{i}]$: The number of activated neighbors of v up to round i where $i = 1..d$.

We store vertices in a (max) priority queue where priority of a vertex v is the sum of $iN_v + aN_v$. At each iteration, we pickup the vertex with highest priority into the spreader set P. The newly selected vertex might cause a chain-reaction activating a sequence of vertices and lower the rounds in which vertices are activated. New activated vertices are successively pushed into the queue Q for further updating much like what happens in the Bellman-Ford shortest-paths algorithm. The algorithm ends when P can activate all vertices in G within d rounds.

Time Complexity. We observe that for each node $t \in V$, changing of r_t can cause at most d update for $rN_w[\cdot]$ when w is a neighbor of t. For all neighbors of t, the total number of update is, hence, $O(d \cdot \text{degree}(t))$. The total time for updating $rN_w\ \forall w \in V$ will be $O((m+n)\cdot d)$ where n, m are the numbers of vertices and edges in G.

Furthermore, we need to extract at most n vertices from the priority queue and adjust values of $iN_u\ \forall u \in V$ no more than $m+n$ times. Since using van Emde Boas tree [41] gives $O(\log\log n)$ performance for all inserts, removals and adjustment, it costs only $O((m+n)\log\log n)$ time for all operations involving in the priority queue.

Summing up, the running time of Algorithm 1 is $O((m+n)(d+\log\log n))$ that is well-scalable even on very large networks of millions vertices.

Algorithm 1 Viral Advertising (VirAds)

```
Input: Undirected graph G = (V,E) and dℕ⁺
Output: A d-Spreader of small size.
Inactive neighbors: iN_v ← d_v ∀v ∈ V
Active neighbors needed: aN_v ← ρ · d_v ∀v ∈ V
Activated rounds: r_v ← d + 1 ∀v ∈ V
Initialize rN_v[i] = 0 ∀i = 0..d
P ← ∅
while there exist inactive vertices do
    u ← argmax_{v∉P} {iN_v + aN_v}
    P ← P ∪ {u}
    if (iN_u + aN_u = 0) then Output P
    Initialize a queue: Q ← {(u, r_v)}
    r_u ← 0
    for all neighbor x of u do
        iN_x ← iN_x − 1
        aN_x ← max{aN_x − 1, 0}
    end for
    while Q ≠ empty do
        (t, oldRound_t) ← Q.pop()
        for all neighbor w of t do
            for i = r_t to min{oldRound_t − 1, r_w − 2} do
                rN_w[i] = rN_w[i] + 1
                if (rN_w[i] ≥ ρ · d_w) then
                    if (r_w ≥ d) ∧ (i + 1 < d)
                        for all neighbor x of w do
                            aN_x ← max{aN_x − 1, 0}
                    r_w = i + 1
                    if w is not activated before
                        for all neighbor x of w do
                            iN_x ← iN_x − 1
                    if (w ∉ Q) then Q.push((w, r_w))
                end if
            end for
        end for
    end while
    Output P
end while
```

3.1.2 Approximating Connected Generalized Dominating Sets

Theorem 9. *There is a polynomial time algorithm that find a weighted C-GDS of size at most* $1.55 \ln \Delta + \frac{5}{2}$ *time that of the minimum C-GDS.*

Proof. We use the same approach for connected dominating problem in [27]. In the first stage, a GDS is found using the greedy heuristic described in Theorem 8. In the second stage, we use a Steiner tree algorithm to connect it.

Since a GDS is also a (normal) dominating set, adding the optimal solution of C-GDS to the found GDS in the first stage gives a Steiner tree of the GDS.

Hence, using a c-approximation algorithm for Steiner tree to connect vertices in a GDS, we obtain a $c\,(H(\Delta)+\ln 2+1)$ approximation algorithm for C-GDS problem. Apply the best known approximation ratio for Steiner tree problem $c \approx 1.55$ by Robins and Zelikovsky [44], the two stages algorithm admits a $1.55\ln\Delta + 2.5$ approximation algorithm. □

Theorem 10. *There are polynomial time algorithms that approximate unweighted C-GDS, TC-GDS within ratios $H(3\Delta)$ and $H(2\Delta)$, respectively.*

Proof. The greedy algorithm presented in [9] can be extended to work with arbitrary threshold function $r_v(x)$ instead of $r_v(x) = \lceil x/2 \rceil$. The analysis involves a non-supmodular potential function that can be overcome with techniques in [45, 46]. Though, it is unclear if the technique can be extended for the weighted cases. □

3.2 Power-law Networks

Many social, biological, and technology networks including OSNs display a non-trivial topological feature: their degree sequences can be well-approximated by a power-law distribution [47]. Many optimization problems that are hard on general graphs, can be solved much more efficient in power-law graphs [48, 49].

We use the well-known $P(\alpha,\beta)$ model[50] in which there are y vertices of degree x, where x and y satisfy $\log y = \alpha - \beta \log x$. In other words,

$$|\{v : d(v) = x\}| = y = \frac{e^{\alpha}}{x^{\beta}}$$

Basically, α is the logarithm of the size of the graph and β is the log-log growth rate of the graph. Graphs with a same β value often express the same behaviors and common characteristics. Hence, it is natural to categorize all graphs sharing a β value into a family of graphs and regard to β as an constant (not a part of the input). Without affecting the conclusions, we will simply use real number instead of rounding down to integers. The error terms can be easily bounded and are sufficiently small in our proofs.

The maximum degree in a $P(\alpha,\beta)$ graph is $e^{\frac{\alpha}{\beta}}$. The number of vertices and edges are

$$n = \sum_{x=1}^{e^{\frac{\alpha}{\beta}}} \frac{e^{\alpha}}{x^{\beta}} \approx \begin{cases} \zeta(\beta)e^{\alpha} & \text{if } \beta > 1 \\ \alpha e^{\alpha} & \text{if } \beta > 1 \\ \frac{e^{\frac{\alpha}{\beta}}}{1-\beta} & \text{if } \beta < 1 \end{cases}, \quad m = \frac{1}{2}\sum_{x=1}^{e^{\frac{\alpha}{\beta}}} x\frac{e^{\alpha}}{x^{\beta}} \approx \begin{cases} \frac{1}{2}\zeta(\beta-1)e^{\alpha} & \text{if } \beta > 2 \\ \frac{1}{4}\alpha e^{\alpha} & \text{if } \beta = 2 \\ \frac{1}{2}\frac{e^{\frac{2\alpha}{\beta}}}{2-\beta} & \text{if } \beta < 2 \end{cases}$$

where $\zeta(\beta) = \sum_{i=1}^{\infty} \frac{1}{i^{\beta}}$ is the Riemann Zeta function.

Theorem 11. *Minimum PIDS is APX-hard i.e. there is a positive constant ε depends only on the log-log growth rate β such that approximating PIDS withiin $1+\varepsilon$ is NP-hard[51].*

Theorem 12. *In a power-law graph $G \in P(\alpha, \beta)$, the size of the optimal PIDS*

$$\mathrm{OPT}_{\mathrm{PIDS}} = \begin{cases} \Omega(n^{\beta}) & \text{if } \beta < 1 \\ \Omega(n/\log n) & \text{if } \beta = 2 \\ \Omega(n) & \text{if } \beta > 2 \end{cases}$$

Proof. 1. Let k be the size of the optimal PIDS. Note that all nodes of degree more than k/ρ must be selected (otherwise the number of selected neighbors will exceed k). The number of nodes with degree larger than k will be

$$k > \sum_{x=k/\rho}^{e^{\frac{\alpha}{\beta}}} \frac{e^{\alpha}}{x^{\beta}} > \sum_{x=k/\rho}^{e^{\frac{\alpha}{\beta}}} \frac{e^{\alpha}}{x} = e^{\alpha}(\ln e^{\frac{\alpha}{\beta}} - \ln k) \tag{5}$$

Solving the above relation gives us $k > \Omega(e^{\alpha}) = \Omega(n^{\beta})$.

2. Similarly, when $\beta = 2$, we have $k = \Omega(e^{\alpha}) = \Omega(n/\log n)$.

3. We use a dual setting approach to obtain the lower bound. Consider the following linear program and its dual of the PIDS problem

$$\begin{aligned} &\textbf{LP}: \min \sum_{v\in V} x_v && \textbf{DP}: \max \sum_{u\in V} r_u y_u - \sum_{v\in V} z_v \\ &\text{s. t. } r_v x_v + \sum_{u\in N(v)} x_u \geq r_v && \text{s. t. } r_u y_u + \sum_{v\in N(u)} y_v - z_u \leq 1 \\ &-x_u \geq -1 && z_v \geq 0 \\ &x_u \geq 0 && y_v \geq 0 \end{aligned} \tag{6}$$

where $r_u = \rho d_u$. We note that for the integral versions of LP(6) and DP(6), both setting $r_u = \rho d_u$ and $r_u = \lceil \rho d_u \rceil$ yield the same optimal solutions, however, setting $r_u = \rho d_u$ simplifies the approximation ratio analysis.

Set $y_u = \gamma\, \forall u \in V$. We solve for value of γ to achieve the tightest lower bound on the size of the optimal PIDS.

To satisfy constraints in the dual, set $z_u = \max\{(\rho+1)d_u\gamma - 1, 0\}$. Then the objective value becomes

$$DP = \rho\gamma \sum_{u\in V} d_u - \sum_{u\in V} \max\{(\rho+1)d_u\gamma - 1, 0\} \tag{7}$$

$$= \rho\gamma \sum_{u\in V} d_u - \sum_{d_u > \tau(\gamma)} ((\rho+1)d_u\gamma - 1) \tag{8}$$

where $\tau(\gamma)$ denotes $(\rho+1)^{-1}\gamma^{-1}$.

Substitute $\gamma = \frac{1}{(\rho+1)\tau}$ into (8), we obtain

$$DP = \frac{\rho}{(\rho+1)\tau} \sum_{x=1}^{e^{\frac{\alpha}{\beta}}} \frac{e^{\alpha}}{x^{\beta}} x - \sum_{x>\tau} \left(\frac{1}{\tau}\frac{e^{\alpha}}{x^{\beta}} x - \frac{e^{\alpha}}{x^{\beta}}\right) \tag{9}$$

$$= \frac{\rho\ \zeta(\beta-1)}{(\rho+1)\tau} e^{\alpha} - \sum_{x>\tau} \left(\frac{1}{\tau}\frac{e^{\alpha}}{x^{\beta-1}} - \frac{e^{\alpha}}{x^{\beta}}\right) \tag{10}$$

Except for at most $\lfloor e^{\frac{\alpha}{\beta}} \rfloor$ points $\tau = 1, 2, \ldots, \lfloor e^{\frac{\alpha}{\beta}} \rfloor$, the derivatives of the objective function, $\frac{\mathrm{d}DP}{\mathrm{d}\tau}$, is defined. Moreover, at those integral points, both one-sided limits, $\lim_{\tau \to i^-} DP$ and $\lim_{\tau \to i^+} DP$, agree i.e. DP is a continuous function everywhere with respect to τ.

Lemma 4. *For every $\tau \in (i, i+1), i \in \mathbb{N}^+$, the derivative $\frac{\mathrm{d}DP}{d\tau}$ is defined and satisfies*

$$\frac{\mathrm{d}DP}{\mathrm{d}\tau} = -\frac{1}{\tau^2} \left(\frac{\rho\ \zeta(\beta-1)}{\rho+1} e^{\alpha} - \sum_{x>\tau} \frac{e^{\alpha}}{x^{\beta-1}} \right) \tag{11}$$

By (11), there exists an fixed dividing point $x_0 \in \mathbb{N}^+$ that depends only on β, satisfying $\frac{\mathrm{d}DP}{\mathrm{d}\tau}(\tau) \geq 0,\ \forall \tau < x_0$ and $\frac{\mathrm{d}DP}{\mathrm{d}\tau}(\tau) < 0,\ \forall \tau > x_0$. Since DP is continuous everywhere, it obtains the global maximum value at $\tau = x_0$.

We show that the value of DP at $\tau = x_0$ is $\Omega(n)$, and since the objective of the primal is lower bounded by DP, it follows that the size of the minimum PIDS will be at least $\Omega(n)$.

$$DP(x_0) = \frac{1}{x_0} \left(\frac{\rho\ \zeta(\beta-1)}{(\rho+1)} e^{\alpha} - \sum_{x>x_0} \frac{e^{\alpha}}{x^{\beta-1}} \right) + \sum_{x>x_0} \frac{e^{\alpha}}{x^{\beta}}$$
$$\geq \sum_{x>x_0} \frac{e^{\alpha}}{x^{\beta}} \approx \left(\zeta(\beta) - \sum_{x \leq x_0} \frac{1}{x^{\beta}}\right) e^{\alpha} \approx \left(1 - \frac{\sum_{x \leq x_0} \frac{1}{x^{\beta}}}{\zeta(\beta)}\right) n = \Omega(n) \quad \square$$

Using the same approach in Theorem 12, we have similar bounds for T-PIDS.

Theorem 13. *In a power-law graph $G \in P(\alpha, \beta)$, the size of the optimal T-PIDS,*

$$\mathrm{OPT}_{\mathrm{T-PIDS}} = \begin{cases} \Omega(n) & \text{if } \beta < 1 \text{ or } \beta > 2 \\ \Omega(n/\log n) & \text{if } \beta = 1 \end{cases}$$

Theorem 14. *In a power-law graph $G \in P(\alpha, \beta)$, the size of the optimal C-PIDS,*

$$\mathrm{OPT}_{\mathrm{C-PIDS}} = \begin{cases} \Omega(n) & \text{if } \beta > 2 \\ \Omega(n/\log n) & \text{if } \beta = 1 \end{cases}$$

If networks have optimal PIDS/T-PIDS/C-PIDS of $\Omega(n)$ size, clearly, any algorithms that produce valid PIDS/T-PIDS/C-PIDS will be constant factor approximation algorithms.

Theorem 15. *Given a power-law $P(\alpha, \beta)$ graph $G = (V, E)$*

1. *$\beta < 1$: PIDS is not in APX (cannot be approximated within a constant factor), while T-PIDS admits a constant factor approximation algorithm.*
2. *$\beta > 2$: There exist constant factor approximation algorithms for PIDS, T-PIDS, and C-PIDS problems.*

3.3 Dense Graphs

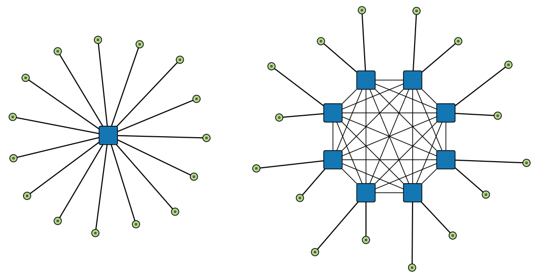

Fig. 3: A PIDS(left) may consist of only one node, while a T-PIDS(right) must contain at least $O(\sqrt{|V|+|E|})$.

Lemma 5. *If $\mathscr{P}_T$ is a T-PIDS of $G=(V,E)$, then $|\mathscr{P}_T| \geq \Omega(\sqrt{|V|+|E|})$.*

Proof. Let $k = |\mathscr{P}_T|$ be the size of an T-PIDS. All $v \in V \setminus \mathscr{P}_T$ must be adjacent to at least one vertex in $\mathscr{P}_T$. Thus, $|V| \leq |\mathscr{P}_T| + \bigcup_{v\in\mathscr{P}_T} |N(v) \setminus \mathscr{P}_T|$. Moreover, for each vertex $v \in \mathscr{P}_T, |N(v) \cap \mathscr{P}_T| \geq \rho|N(v)| \Rightarrow |N(v) \setminus \mathscr{P}_T| \leq \frac{1-\rho}{\rho}|N(v) \cap \mathscr{P}_T| \leq \frac{1-\rho}{\rho}(k-1)$. Therefore

$$n \leq k + k\frac{1-\rho}{\rho}(k-1) = \frac{1-\rho}{\rho}k^2 + \frac{2\rho-1}{\rho}k \tag{12}$$

Divide edges in E into three categories: (1) edges whose both ends are in $\mathscr{P}_T$, (2) edges whose exact one end is in $\mathscr{P}_T$, (3) edges whose both ends are not in $\mathscr{P}_T$. We have at most $\binom{k}{2}$ edges of type 1. For a vertex $v \in \mathscr{P}_T$, the number of type 2 edges incident to v is at most $\frac{1-\rho}{\rho}(k-1)$ since v is adjacent to at most $k-1$ vertices in $\mathscr{P}_T$. Hence, the number of type 2 edges is upper bounded by $2k\frac{1-\rho}{\rho}(k-1) = \frac{2(1-\rho)}{\rho}\binom{k}{2}$. For each vertex $u \notin \mathscr{P}_T$, the number of type 3 edges incident to u is at most $\frac{1-\rho}{\rho}$ times the number of type 2 edges incident to u. Therefore, the number of type 3 edges is at most $\frac{(1-\rho)^2}{\rho^2}\binom{k}{2}$.

Adding all three types of edges together, we have

$$|E| \leq \left(1 + \frac{2(1-\rho)}{\rho} + \frac{(1-\rho)^2}{\rho^2}\right)\binom{k}{2} = \frac{1}{\rho^2}\binom{k}{2} \tag{13}$$

It follows from (12) and (13) that $|\mathscr{P}_T| = k = \Omega(\sqrt{|V|+|E|})$. $\square$

The bound is tight i.e. we can construct a T-PIDS of size $\Omega(\sqrt{|V|+|E|})$. For example we construct a 'hairy' clique of $n = k + k \cdot \lfloor \frac{1-\rho}{\rho}(k-1) \rfloor$ vertices and $m = \binom{k}{2} + k \cdot \lfloor \frac{1-\rho}{\rho}(k-1) \rfloor$ edges by connecting each vertex in a clique of size k to $\lfloor \frac{1-\rho}{\rho}(k-1) \rfloor$ leaf nodes (Fig. 3). The minimum T-PIDS will be the clique itself that is of size $k = \Omega(\sqrt{n+m})$.

Theorem 16. *For a dense graph $G = (V,E)$ with $|E| = \Omega(|V|^2)$, there exist constant approximation algorithms for PIDS, T-PIDS, and C-PIDS problems.*

3.4 Finding Optimal Solutions in Trees

In trees, it is possible to find optimal GDS and T-GDS in polynomial time. However, designing such algorithms in a linear-time fashion is not too obvious. We present two Depth-first search-based (DFS) algorithms in Algorithms 2 and 3 for GDS and T-GDS, respectively. Notice that the solution for kC-GDS and TC-GDS problems on trees is trivial; the optimal solution is simply the set of all non-leaf nodes.

GDS-TREE(G)
$\mathscr{P} = \emptyset$
PIDS-VISIT(u), for any $u \in V$
return $\mathscr{P}$

GDS-VISIT(u)
for each unvisited $v \in N(u)$ **do**
    GDS-VISIT(v)
    if $r_p(v,\mathscr{P}) > 0$ **then**
        $\mathscr{P} = \mathscr{P} \cup \{u\}$
if $r_p(u,\mathscr{P}) > 1$ **then**
    $\mathscr{P} = \mathscr{P} \cup \{u\}$

Algorithm 2: GDS-TREE(G)

T-GDS-TREE(G)
$\mathscr{T} = \emptyset$
T-GDS-VISIT(u,u), for any $u \in V$
return $\mathscr{T}$

T-GDS-VISIT(u, p_u)
for each unvisited $v \in N(u)$ **do**
    T-GDS-VISIT(v,u)
if $r_t(u,\mathscr{T}) > 0$ **then**
    $\mathscr{T} = \mathscr{T} \cup \{p_u\}$
Select arbitrary $r_t(u,\mathscr{T})$ unselected neighbors(children) of u into $\mathscr{T}$.

Algorithm 3: T-GDS-TREE(G)

Theorem 17. *Optimal GDS and T-GDS in trees can be found in linear-time.*

Proof. At a given step, $\mathscr{P}/\mathscr{T}$ denote the current GDS/T-GDS. For each $v \in V$, define the functions $r_p(v,\mathscr{P}) = \lceil r_v(d(v)) \rceil (1 - \mathbf{1}_{\mathscr{P}}(v)) - |N(v) \cap \mathscr{P}|$ and $r_t(v,\mathscr{T}) = \lceil r_v(d(v)) \rceil - |N(v) \cap \mathscr{T}|$, where $\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$

Functions $r_p(v,\mathscr{P}), r_t(v,\mathscr{T})$ determine the minimum numbers of v's neighbors to include into the optimal solutions. A node u with $r_p(u,\mathscr{P}) > 0$ or $r_t(u,\mathscr{T}) > 0$ is called *uncovered*, otherwise u is called *covered*.

Assume that the tree is rooted at some vertex u. For an edge (u,v), if u is visited before v, then u is the parent of v and v is a child of u.

Correctness. We show by induction that each selection step is optimal.

GDS: Assume that all the selections made so far are optimal. Assume node u, the parent of v is selected in step 3, Alg. 2. From $r_p(v,\mathscr{P}) > 0$, we have

1. $v \notin \mathscr{P}$ or else $r_p(v,\mathscr{P}) < 0$ and
2. $r_p(v,\mathscr{P}) = 1$ if not v has already been selected by the end of GDS-VISIT(v)).

To cover v, we have to either select v, u or some children of v. However, since all nodes in the subtree rooted at v have been covered. There will be no extra benefit in selecting v or its children. Formally, if we have an optimal solution that selects v or its children, we can always replace the selected vertex with u and obtain a new optimal PIDS. In case $r_p(u,\mathscr{P}) > 1$, we are forced to select u.

T-GDS: After T-GDS-VISIT(v, p_v) finishes, v always becomes covered. Assume the selection of vertices into $\mathscr{T}$ is optimal so far. During the visit of a node u, if $r_t(u,\mathscr{T}) > 0$, we select p_u, the parent of u, if $p_u \notin \mathscr{T}$. Since p_u might cover other vertices, while selecting children of u will not affect any uncovered vertices other than u. Finally, we might have select children of u to fully covered u (but only after p_u is selected).

Time complexity. Values of $r_p(v,\mathscr{P})$ and $r_t(v,\mathscr{T})$ can be maintained in $O(|V|)$. Only when a new vertex is added, we need to update $r_p(.)$ and/or $r_t(.)$ values of that node and all its neighbors. Each node is added at most once, hence the total cost has the same order with the total degree of all vertices i.e. $2|V|$. Adding the time $O(|V|)$ taken by the DFS traversal, the overall time complexities are still $O(|V|)$. □

References

1. Ivan S. Dominating set based bluetooth scatternet formation with localized maintenance. *Parallel and Distributed Processing Symposium, International*, 2002.
2. T. Moscibroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *IEEE International Parallel and Distributed Processing Symposium '05*, IPDPS '05, Washington, DC, USA, 2005. IEEE Computer Society.
3. J. Blum, M. Ding, A. Thaeler, and X. Cheng. Connected dominating set in sensor networks and manets. In Ding-Zhu Du and Panos M. Pardalos, editors, *Handbook of Combinatorial Optimization*, pages 329–369. Springer US, 2005.
4. M.T. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du. Connected dominating sets in wireless networks with different transmission ranges. *Mobile Computing, IEEE Transactions on*, 6(7):721 –730, july 2007.
5. M. T. Thai, R. Tiwari, and D.-Z. Du. On construction of virtual backbone in wireless ad hoc networks with unidirectional links. *IEEE Transactions on Mobile Computing*, 2008.
6. R. Tiwari, T. N. Dinh, and M. T. Thai. On approximation algorithms for interference-aware broadcast scheduling in 2d and 3d wireless sensor networks. In *International Conference on*

Wireless Algorithms, Systems, and Applications '09, WASA '09, Berlin, Heidelberg, 2009. Springer-Verlag.
7. F. Wang, M.T. Thai, and D.-Z. Du. On the construction of 2-connected virtual backbone in wireless networks. *Wireless Communications, IEEE Transactions on*, 8(3):1230 –1237, 2009.
8. F. Wang, E. Camacho, and K. Xu. Positive influence dominating set in online social networks. In *Proceedings of the 3rd International Conference on Combinatorial Optimization and Applications*, COCOA '09, pages 313–321, Berlin, Heidelberg, 2009. Springer-Verlag.
9. X. Zhu, J. Yu, W. Lee, D. Kim, S. Shan, and D.-Z. Du. New dominating sets in social networks. *Journal of Global Optimization*, 48:633–642, 2010. 10.1007/s10898-009-9511-2.
10. Feng Wang, Hongwei Du, Erika Camacho, Kuai Xu, Wonjun Lee, Yan Shi, and Shan Shan. On positive influence dominating sets in social networks. *Theoretical Computer Science*, 412(3):265 – 269, 2011. Combinatorial Optimization and Applications - COCOA 2009.
11. K. G. Hill, J. D. Hawkins, R. F. Catalano, R. D. Abbott, and J. Guo. Family influences on the risk of daily smoking initiation. *Journal of Adolescent Health*, 37(3):202 – 210, 2005.
12. J. B. Standridge, R. G. Zylstra, and S. M. Adams. Alcohol consumption: An overview of benefits and risks. *Southern Medical Journal*, 2004.
13. R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Information Processing Letters*, 89:75–83, January 2004.
14. Bin Wang and Kai-Nan Xiang. On k-tuple domination of random graphs. *Applied Mathematics Letters*, 22(10):1513 – 1517, 2009.
15. S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-[epsilon]. *Journal of Computer and System Sciences*, 2008. Computational Complexity 2003.
16. U. Feige. A threshold of ln n for approximating set cover. *Journal of ACM*, 45(4):634–652, 1998.
17. Aravind Srinivasan. Improved approximations of packing and covering problems. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, STOC '95, pages 268–276, New York, NY, USA, 1995. ACM.
18. B. S. Baker. Approximation algorithms for np-complete problems on planar graphs. *Journal of ACM*, 41:153–180, January 1994.
19. Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of ptas. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 329–337, New York, NY, USA, 1996. ACM.
20. David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, pages 275–291, 2000.
21. Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23:613–632, December 2003.
22. E. D. Demaine and M. Hajiaghayi. Bidimensionality: new connections between fpt algorithms and ptass. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '05, pages 590–601, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
23. M. Gibson and I. Pirwani. Algorithms for dominating set in disk graphs: Breaking the log n barrier. In Mark de Berg and Ulrich Meyer, editors, *European Symposia on Algorithms 2010*, volume 6346 of *Lecture Notes in Computer Science*, pages 243–254. Springer Berlin Heidelberg, 2010.
24. A. Czygrinow, M. Hanckowiak, and W. Wawrzyniak. Fast distributed approximations in planar graphs. In Gadi Taubenfeld, editor, *Distributed Computing*, volume 5218 of *Lecture Notes in Computer Science*, pages 78–92. Springer Berlin / Heidelberg, 2008.
25. D. Dai and C. Yu. A 5+[epsilon]-approximation algorithm for minimum weighted dominating set in unit disk graph. *Theoretical Computer Science*, 410(8-10):756 – 765, 2009.
26. Michael A. Henning. A survey of selected recent results on total domination in graphs. *Discrete Mathematics*, 309(1):32 – 63, 2009.
27. S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20:374–387, April 1998.
28. C. Liao and G. J. Chang. Algorithmic aspect of k-tuple domination in graphs. *Taiwanese J. Math*, 6:415–420, 2003.

29. W. Shang, P. Wan, F. Yao, and X. Hu. Algorithms for minimum m-connected k-tuple dominating set problem. *Theoretical Computer Science*, 381(1-3):241–247, August 2007.
30. N Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Transactions on Algorithms*, 2(2):153–177, 2006.
31. P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of The 7th ACM SIGKDD International Conference on Knowledge Discovery and Datamining*, pages 57–66, New York, NY, USA, 2001. ACM.
32. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD'03: Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM New York, NY, USA, 2003.
33. D. Kempe, J. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages and Programming '05*, pages 1127–1138, 2005.
34. J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining '07*, pages 420–429, New York, NY, USA, 2007. ACM.
35. Feng Z., Zhao Z., and Weili W. Latency-bounded minimum influential node selection in social networks. In Benyuan Liu, Azer Bestavros, Ding-Zhu Du, and Jie Wang, editors, *Wireless Algorithms, Systems, and Applications*, Lecture Notes in Computer Science, pages 519–526, 2009.
36. W. Zhang, Z. Zhang, W. Wang, F. Zou, and W. Lee. Polynomial time approximation scheme for t-latency bounded information propagation problem inwirelessnetworks. *Journal of Combinatorial Optimization*, pages 1–11, 2010. 10.1007/s10878-010-9359-x.
37. M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008.
38. L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *ACM Symposium on Theory of Computing '01*, pages 453–461, New York, NY, USA, 2001. ACM.
39. N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of ACM*, 42:1091–1106, September 1995.
40. S. Rajagopalan and V. V. Vazirani. Primal-dual rnc approximation algorithms for (multi)-set (multi)-cover and covering integer programs. In *Annual IEEE Symposium on Foundations of Computer Science '93*, pages 322–331, Washington, DC, USA, 1993. IEEE Computer Society.
41. Peter van E. B., R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Mathematical Systems Theory*, 10:99–127, 1977.
42. D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. In *SIROCCO'96: Colloquium on Structural Information and Communication Complexity*, pages 152–169, 1996.
43. M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World Wide Web*, WWW '09, pages 721–730, New York, NY, USA, 2009. ACM.
44. G. Robins and A. Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal of Discrete Mathematics*, 19:122–134, May 2005.
45. L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K. Ko. A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science*, 329:325–330, December 2004.
46. D.-Z. Du, R. L. Graham, P. M. Pardalos, P.-J. Wan, W. Wu, and W. Zhao. Analysis of greedy approximations with nonsubmodular potential functions. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 167–175, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.
47. M. Girvan and M. E. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, June 2002.
48. C. Gkantsidis, M. Mihail, and A. Saberi. Conductance and congestion in power law graphs. In *SIGMETRICS '03: Proceedings of the International Conference on Measurements and Modeling of Computer Systems*, pages 148–159, New York, NY, USA, 2003. ACM.

49. A. Ferrante, G. Pandurangan, and K. Park. On the hardness of optimization in power-law graphs. *Theoretical Computer Science*, 393(1-3):220–230, 2008.
50. W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Math*, 10:53–66, 2000.
51. Y. Shen, D. T. Nguyen, Y. Xuan, and M. T. Thai. New techniques for approximating optimal substructure problems in power-law graphs. Theorical Computer Science, 2011.