Connected Domatic Packings in Node-capacitated Graphs

Alina Ene[*]   Nitish Korula[†]   Ali Vakilian[‡]

July 22, 2013

Abstract

A set of vertices in a graph is a dominating set if every vertex outside the set has a neighbor in the set. A dominating set is connected if the subgraph induced by its vertices is connected. The connected domatic partition problem asks for a partition of the nodes into connected dominating sets. The connected domatic number of a graph is the size of a largest connected domatic partition and it is a well-studied graph parameter with applications in the design of wireless networks. In this note, we consider the fractional counterpart of the connected domatic partition problem in *node-capacitated* graphs. Let n be the number of nodes in the graph and let k be the minimum capacity of a node separator in G. Fractionally we can pack at most k connected dominating sets subject to the capacities on the nodes, and our algorithms construct packings whose sizes are proportional to k. Some of our main contributions are the following:

- An algorithm for constructing a fractional connected domatic packing of size $\Omega(k)$ for node-capacitated planar and minor-closed families of graphs.

- An algorithm for constructing a fractional connected domatic packing of size $\Omega(k/\ln n)$ for node-capacitated general graphs.

1 Introduction

Let $G = (V, E)$ be an undirected and connected graph with n nodes. A set S of nodes is a dominating set if every node not in S has a neighbor in S. A connected domatic partition is a collection of connected dominating sets that are node disjoint. The connected domatic number is the size of a largest connected domatic partition. In this note, we consider the problem of constructing large fractional connected domatic packings; a fractional packing is a weight function on connected dominating sets such that, for each vertex v, the total weight of the connected dominating sets that contain v is at most one.

Connected domatic partitions and packings have several applications in the design of wireless networks. In these applications, a connected dominating set is used as a virtual backbone, and the rest of the nodes use the connected dominating set to exchange messages and route traffic [6, 7, 21]. Motivated by the goal of improving the energy efficiency and the lifetime of the network, several papers [22–24] have proposed using several connected dominating sets; these approaches first compute a large connected domatic packing or partition and they rotate between the connected dominating sets. Additionally, the recent work of Censor-Hillel *et al.* [5] establishes a close connection between the fractional connected domatic number and the throughput of store-and-forward algorithms for routing in wireless networks.

Integer and fractional packings of combinatorial structures are connected to each other and to the corresponding optimization problem that asks for the minimum cost combinatorial structure; we refer the reader to Section 5 in [3] for an overview of these connections. In particular, an α-approximation for the minimum-cost Connected Dominating Set (Min-Cost-CDS) problem implies an α-approximation for the Connected Domatic Packing (CDS-Packing) problem; this connection was shown by Carr and Vempala [4]. This result and the $O(\ln n)$ approximation algorithm for Min-Cost-CDS given by Guha and Khuller [11] imply an $O(\ln n)$ approximation for CDS-Packing. In very recent work, Censor-Hillel *et al.* [5] gave the first poly-logarithmic approximation for the Connected Domatic Partition problem; their algorithm achieves an $O(\ln^5 n)$ approximation. The results of [5] guarantee partitions and packings whose sizes are a poly-logarithmic fraction of the *vertex connectivity*[1]. These guarantees are independent of the size of the largest partition or packing and thus they are not approximation results *per se*. Since the connectivity of the graph is an upper bound on the fractional connected domatic number and thus the connected domatic number as well, these absolute results give us approximation guarantees as a byproduct.

In several applications in wireless networks, each node has a certain battery life that constrains how long the node can be used as part of a virtual backbone for the network. We can model such networks using node-capacitated graphs, where the capacity represents the battery life of the node. Motivated in part by these applications, we consider the more general problem of constructing large connected packings in node-capacitated graphs. In this setting, each vertex v has a capacity $\mathrm{cap}(v)$ and the goal is to find a fractional packing of maximum total weight such that the fractional weight of the connected dominating sets that contain each vertex is at most the capacity of the vertex. We refer to the capacitated analogue of CDS-Packing as Cap-CDS-Packing. We can reduce the capacitated problem to the uncapacitated one by replacing each node by a clique whose size is equal to the capacity of the node. However, this reduction does not run in polynomial time if the capacities are large and it does not preserve the special structure of certain graphs, such as planar or minor-free graphs. Real-world wireless networks are typically not arbitrary graphs but rather they are nearly planar or have restricted structure. We give an algorithm that constructs improved fractional packings for such networks.

Theorem 1.1. *Let G be a node-capacitated graph that belongs to a minor-closed family $\mathcal{G}$ of graphs.*

---

[1] A graph $G = (V, E)$ is k-vertex-connected iff, for any subset $S \subseteq V$ of size less than k, the removal of S does not disconnect the graph. The vertex connectivity of G is the maximum k such that G is k-vertex-connected.

Let k be the minimum capacity of a node separator[2] in G. There is a polynomial time algorithm that constructs a fractional connected domatic packing in G of size $\Omega(k)$, where the constant depends only on the family $\mathcal{G}$.

Our approach can also be used to construct fractional packings for general graphs with arbitrary node capacities. This result was shown in [5] for uncapacitated graphs using very different techniques.

Theorem 1.2. *Let G be a node-capacitated graph. Let k be the minimum capacity of a node separator in G. There is a polynomial time algorithm that constructs a fractional connected domatic packing in G of size $\Omega(k/\ln n)$.*

Our algorithm for Cap-CDS-Packing is based on a connection between the size of a fractional packing and the integrality gap of a standard LP relaxation for Min-Cost-CDS; we describe this LP relaxation in Section 2. We show that, if the relaxation has an integrality gap of r, we can construct a packing of size k/r in polynomial time using an r-approximate rounding algorithm for the Min-Cost-CDS LP and the ellipsoid method. One of our contributions is a constant upper bound on the integrality gap of Min-Cost-CDS LP in minor-closed families of graphs, where the constant depends only on the family. In the process, we also show that the integrality gap of a standard LP relaxation for the minimum cost Dominating Set (Min-Cost-DS) problem is constant in minor-free graphs. The Min-Cost-DS problem admits a PTAS in planar graphs [1], but this result does not establish an upper bound on the integrality gap. Our algorithms can be easily adapted to give analogous integrality gap upper bounds for the Steiner variants of Min-Cost-DS and Min-Cost-CDS; in the Steiner problems, we are given a subset of the vertices called the terminals and the goal is to select a (connected) set that dominates the terminals.

Theorem 1.3. *The standard LP relaxation for the Min-Cost-DS problem has an $O(1)$ integrality gap in planar and minor-closed families of graphs. Moreover, there is a polynomial time algorithm that rounds any fractional solution to an integral solution whose cost is at most $O(1)$ times larger than the cost of the fractional solution.*

Theorem 1.4. *The standard LP relaxation for the Min-Cost-CDS problem has an $O(1)$ integrality gap in planar and minor-closed families of graphs. Moreover, there is a polynomial time algorithm that rounds any fractional solution to an integral solution whose cost is at most $O(1)$ times larger than the cost of the fractional solution.*

Other related work: Domatic partitions have received considerable attention; we refer the reader to [13–15] for a comprehensive treatment of graph domination. Feige *et al.* [9] gave a polynomial time algorithm that constructs a domatic partition of size $\Omega(\delta/\ln n)$, where δ is the minimum degree of the graph and they showed that this is best possible unless $\mathrm{NP} \subseteq \mathrm{DTIME}\left(n^{\log\log n}\right)$. Călinescu *et al.* [3] considered the more general problem of packing disjoint bases in a polymatroid.

2   Algorithm for fractional connected domatic packings

In this section, we give polynomial time algorithms for constructing fractional connected domatic packings in node-capacitated graphs.

We start by introducing the following natural LP relaxation for the Min-Cost-CDS problem. Let $G = (V, E)$ be a graph with costs $\text{cost}(v)$ associated with the nodes. For each vertex v, we let $\Gamma(v)$

---

[2]A set S is a node separator in G if the graph $G - S$ has at least two connected components, where $G - S$ is the graph obtained from G by removing the nodes of S

denote the set of all neighbors of v in G. For a set S of nodes, we let $\Gamma(S)$ denote the set of all nodes v such that v is not in S and v has a neighbor in S. Let $\Gamma^+(v) = \Gamma(v) \cup \{v\}$. The relaxation has a variable $x(v)$ for each vertex v with the interpretation that $x(v) = 1$ iff v is in the connected dominating set. Let $\mathcal{S}$ be the collection of all sets S such that S, $\Gamma(S)$, and $V - (S \cup \Gamma(S))$ are all non-empty; note that, for each set $S \in \mathcal{S}$, the set $\Gamma(S)$ is a node separator that separates S from $V - (S \cup \Gamma(S))$. The relaxation Min-Cost-CDS-LP is given below.

$$
\begin{array}{lll}
& \text{Min-Cost-CDS-LP} & \\
\min & \displaystyle\sum_{v \in V} x(v)\text{cost}(v) & \\
\text{s.t.} & \displaystyle\sum_{u \in \Gamma^+(v)} x(u) \geq 1 & v \in V \\
& \displaystyle\sum_{v \in \Gamma(S)} x(v) \geq 1 & S \in \mathcal{S} \\
& x(v) \geq 0 & v \in V
\end{array}
$$

Note that the LP is a valid relaxation for the Min-Cost-CDS problem. A dominating set must contain a vertex from $\Gamma^+(v)$ for each vertex v. Additionally, a connected dominating set must contain a vertex from each node separator.

The two main steps of our approach for constructing large fractional packings are the following. The first step is to show that we can construct in polynomial time a packing of size $\Omega(k/r)$, where k is the capacity of a minimum node separator in G and r is an upper bound on the integrality gap of Min-Cost-CDS-LP; we refer the reader to Corollary 2.2 for a precise statement of the result. The second step is to upper bound the integrality gap of Min-Cost-CDS-LP. For general graphs, it follows easily from previous work that the integrality gap is $O(\ln n)$ and thus we can find a packing of size $\Omega(k/\ln n)$. For planar graphs and more generally, minor-free graphs, we will show that the integrality gap of Min-Cost-CDS-LP is a constant and thus we can find a packing of size $\Omega(k)$.

In the following, we say that a rounding algorithm $\mathcal{A}$ for an LP relaxation is an r-approximate rounding algorithm for the LP if, given any fractional solution to the LP, the algorithm constructs an integral solution of value at most r times the value of the fractional solution.

Consider an instance $\langle G, \text{cap}\rangle$ of Cap-CDS-Packing, where G is a graph from a family $\mathcal{G}$ of graphs and $\text{cap}(\cdot)$ is a capacity function on the nodes of G. Let k be the minimum capacity of a node separator in G. Our goal is to show that we can construct a fractional packing of size $\Omega(k/r)$ provided that we have an r-approximate rounding algorithm for Min-Cost-CDS-LP. This will follow from the theorem below, which is an immediate corollary of Theorem 2 in [4].

Theorem 2.1 (Carr and Vempala [4]). *Let $\mathcal{G}$ be a family of graphs. Let $\mathbf{x}$ be a fractional solution to Min-Cost-CDS-LP for an instance of Min-Cost-CDS for which the graph G is in $\mathcal{G}$. Let $\mathcal{A}$ be a polynomial time rounding algorithm for Min-Cost-CDS-LP that is r-approximate on instances for which the graph is in $\mathcal{G}$. Given $\mathbf{x}$ and $\mathcal{A}$, we can find in polynomial time a collection of polynomially many connected dominating sets $D_1, \ldots, D_\ell$ with associated weights $\lambda_1, \ldots, \lambda_\ell$ such that $\sum_{i=1}^{\ell} \lambda_i = 1$ and, for each vertex v, we have $\sum_{i : v \in D_i} \lambda_i \leq r \cdot x(v)$.*

The theorem above gives us the following corollary.

Corollary 2.2. *Let $\mathcal{G}$ be a family of graphs. Let $\mathcal{A}$ be a polynomial time rounding algorithm for Min-Cost-CDS-LP that is r-approximate on instances for which the graph is in $\mathcal{G}$. Let $\langle G, \text{cap}\rangle$ be an*

instance of Cap-CDS-Packing *such that* $G \in \mathcal{G}$*. Let* k *be the minimum capacity of any node separator in* G*. Given* $\mathcal{A}$ *and* $\langle G, \mathrm{cap} \rangle$*, we can find in polynomial time a collection of polynomially many connected dominating sets* $D_1, \ldots, D_\ell$ *and associated weights* $\alpha_1, \ldots, \alpha_\ell$ *such that* $\sum_{i=1}^{\ell} \alpha_i \geq k/r$ *and, for each vertex* $v \in V(G)$*, we have* $\sum_{i: v \in D_i} \alpha_i \leq \mathrm{cap}(v)$*. Differently said,* $\{\langle D_1, \alpha_1 \rangle, \ldots, \langle D_\ell, \alpha_\ell \rangle\}$ *is a feasible fractional connected domatic packing of size* $\Omega(k/r)$*.*

Proof: Consider the following fractional solution $\mathbf{x}$: $x(v) = \mathrm{cap}(v)/k$ for each vertex $v \in V(G)$. We can verify that $\mathbf{x}$ is a feasible solution to Min-Cost-CDS-LP as follows. Consider a vertex v. We can assume that $\Gamma(v)$ is a node separator; otherwise, $\Gamma^+(v) = V(G)$ and $\sum_{u \in \Gamma^+(v)} x(u) \geq 1$ trivially holds. Since $\Gamma(v)$ is a node separator, it follows that $\mathrm{cap}(\Gamma(v)) \geq k$. Therefore we have

$$\sum_{u \in \Gamma(v)} x(u) = \frac{1}{k} \sum_{u \in \Gamma(v)} \mathrm{cap}(u) \geq 1,$$

and thus $\mathbf{x}$ satisfies the first set of constraints. Consider a set $S \in \mathcal{S}$. Since $\Gamma(S)$ is a node separator in G it follows that $\mathrm{cap}(\Gamma(S)) \geq k$. Therefore we have

$$\sum_{v \in \Gamma(S)} x(v) = \frac{1}{k} \sum_{v \in \Gamma(S)} \mathrm{cap}(v) \geq 1,$$

and thus $\mathbf{x}$ satisfies the second set of constraints.

We apply Theorem 2.1 to $\mathbf{x}$ and $\mathcal{A}$ in order to get a collection of connected dominating sets $D_1, \ldots, D_\ell$ and associated weights $\lambda_1, \ldots, \lambda_\ell$. For each i, let $\alpha_i = (k/r) \cdot \lambda_i$. We can verify that $\{\langle D_1, \alpha_1 \rangle, \ldots, \langle D_\ell, \alpha_\ell \rangle\}$ is the desired packing as follows. We have

$$\sum_{i=1}^{\ell} \alpha_i = \frac{k}{r} \sum_{i=1}^{\ell} \lambda_i = \frac{k}{r}.$$

Additionally, for each vertex v, we have

$$\sum_{i: v \in D_i} \alpha_i = \frac{k}{r} \sum_{i: v \in D_i} \lambda_i \leq \frac{k}{r} \cdot r \cdot x(v) = \mathrm{cap}(v).$$

$\square$

In the second step, we upper bound the integrality gap of Min-Cost-CDS-LP. To this end, we will first relate the integrality gap of Min-Cost-CDS-LP to the integrality gaps of the standard LP relaxations for the minimum-cost Dominating Set (Min-Cost-DS) problem and the minimum node-weighted Steiner Tree (NW-Steiner-Tree) problem, and then we will upper bound the integrality gaps of these two relaxations. The LP relaxation for Min-Cost-DS is the relaxation Min-Cost-DS-LP given below. In the NW-Steiner-Tree problem, we are given a graph $G = (V, E)$ with non-negative weights $w(v)$ on the nodes and a set $T \subseteq V$ of nodes called *terminals*. The goal is to select a minimum weight subgraph H of G that spans all the terminals, where the weight of H is the total weight of the nodes in H. (Note that we may assume that H is a node-induced connected subgraph of G.) Let $\mathcal{S}_T$ be the collection consisting of all sets S such that S separates the terminals; more precisely, $S \cap T$ and $(V - S) \cap T$ are both non-empty. For each set $S \in \mathcal{S}_T$, at least one vertex in $\Gamma(S)$ must be in the solution. The LP relaxation for NW-Steiner-Tree is the relaxation NW-Steiner-Tree-LP given below.

Min-Cost-DS-LP	NW-Steiner-Tree-LP
$\min \quad \sum_{v \in V} x(v) \text{cost}(v)$ $\text{s.t.} \quad \sum_{u \in \Gamma^+(v)} x(u) \geq 1 \qquad v \in V$ $x(v) \geq 0 \qquad v \in V$	$\min \quad \sum_{v \in V} x(v) w(v)$ $\text{s.t.} \quad \sum_{v \in \Gamma(S)} x(v) \geq 1 \qquad S \in \mathcal{S}_T$ $x(v) \geq 0 \qquad v \in V$

The following straightforward propositions allow us to relate the integrality gap of Min-Cost-CDS-LP to the integrality gaps of Min-Cost-DS-LP and NW-Steiner-Tree-LP.

Proposition 2.3. *Consider an instance of* Min-Cost-CDS*; let G be the input graph and let $\mathbf{x}$ be a feasible solution to* Min-Cost-CDS-LP *for this instance. Then $\mathbf{x}$ is a feasible solution to* Min-Cost-DS-LP *for any instance of* Min-Cost-DS *in which the input graph is G.*

Proposition 2.4. *Consider an instance of* Min-Cost-CDS*; let G be the input graph and let $\mathbf{x}$ be a feasible solution to* Min-Cost-CDS-LP *for this instance. Let T be any subset of the vertices and let $\mathbf{x}'$ be the following fractional solution: $x'(v) = x(v)$ if $v \notin T$ and $x'(v) = 1$ otherwise. Then $\mathbf{x}'$ is a feasible solution to* NW-Steiner-Tree-LP *for any instance of* NW-Steiner-Tree *in which the input graph is G and the set of terminals is T.*

Proof: Consider a set $S \in \mathcal{S}_T$. If $V - (S \cup \Gamma(S))$ is non-empty, $S \in \mathcal{S}$ and the fact that $\mathbf{x}$ is a feasible solution to Min-Cost-CDS-LP gives us that $x(\Gamma(S))$ is at least one. Therefore we may assume that $S \cup \Gamma(S) = V$. Since S separates the terminals, $\Gamma(S)$ contains a terminal and thus $x'(\Gamma(S))$ is at least one. $\qquad\square$

Corollary 2.5. *Let $\mathcal{G}$ be a family of graphs. Let $\mathcal{A}_1$ be a polynomial time rounding algorithm for* Min-Cost-DS-LP *that is r_1-approximate on instances in which the graph is in $\mathcal{G}$. Let $\mathcal{A}_2$ be a polynomial time rounding algorithm for* NW-Steiner-Tree-LP *that is r_2-approximate on instances in which the graph is in $\mathcal{G}$. Given $\mathcal{A}_1$ and $\mathcal{A}_2$, we can design a polynomial time a rounding algorithm for* Min-Cost-CDS-LP *that is $(r_1 + r_2)$-approximate on instances in which the graph is in $\mathcal{G}$.*

Proof: Let $\langle G, \text{cost} \rangle$ be an instance of Min-Cost-CDS, where $G \in \mathcal{G}$. Let $\mathbf{x}$ be a feasible solution to Min-Cost-CDS-LP for this instance. Let $C = \sum_{v \in V} x(v) \text{cost}(v)$. Our goal is to show that we can construct in polynomial time a connected dominating set D' whose cost $\text{cost}(D')$ is at most $(r_1 + r_2)C$. By Proposition 2.3, $\mathbf{x}$ is a feasible solution to Min-Cost-DS-LP for the instance $\langle G, \text{cost} \rangle$. Thus we can run $\mathcal{A}_1$ with $\mathbf{x}$ as input in order to get a dominating set D such that $\text{cost}(D) \leq r_1 \cdot C$. Once we have the dominating set D, we consider the following instance of NW-Steiner-Tree. The nodes in D will be the terminals. We define a set of weights as follows: for each vertex v, we have $w(v) = \text{cost}(v)$ if $v \notin D$ and $w(v) = 0$ otherwise. We define a fractional solution $\mathbf{x}'$ as follows: for each vertex v, we have $\mathbf{x}'(v) = \mathbf{x}(v)$ if $v \notin D$ and $\mathbf{x}'(v) = 1$ otherwise. Let $W = \sum_{v \in V} w(v) x'(v)$; note that $W = \sum_{v \in V - D} x(v) \text{cost}(v) \leq C$. By Proposition 2.4, $\mathbf{x}'$ is a feasible solution to NW-Steiner-Tree-LP for the instance $\langle G, w, D \rangle$. Thus we can run $\mathcal{A}_2$ with $\mathbf{x}'$ as input in order to get a node-induced connected subgraph H of G that spans D and it has weight $w(H) \leq r_2 \cdot W$. Let $D' = V(H)$; since D is a subset of D', D' is a dominating set. Additionally, $\text{cost}(D') \leq (r_1 + r_2)C$. $\qquad\square$

Consider the relaxation Min-Cost-DS-LP. For general graphs, we can show an $O(\ln n)$ upper bound on the integrality gap using the following standard randomized rounding approach. Given a fractional solution $\mathbf{x}$, we select a set D of nodes as follows: for each vertex v, we add v to D independently at random with probability $\min\{c \ln n \cdot x(v), 1\}$, where c is a large enough constant. With high probability, the resulting set D is a dominating set. For minor-closed families of graphs, we give a primal-dual

algorithm in Section 3 that shows that the integrality gap is $O(1)$. We remark that the Min-Cost-DS problem admits a PTAS in planar graphs [1], but the algorithm of [1] does not give an upper bound on the integrality gap of the LP.

Theorem 2.6. *Let $\mathcal{G}$ be a minor-closed family of graphs. There is a polynomial time rounding algorithm for* Min-Cost-DS-LP *that is $c(\mathcal{G})$-approximate on instances in which the graph is in $\mathcal{G}$, where $c(\mathcal{G})$ is a constant that depends only on the family $\mathcal{G}$.*

Finally, consider the relaxation NW-Steiner-Tree. Guha *et al.* [12] showed that the integrality gap is $O(\ln n)$ for general graphs, and Demaine *et al.* [8] showed that the integrality gap is $O(1)$ for minor-closed families of graphs. This completes the proof of Theorem 1.2 and Theorem 1.1.

3   Algorithm for Min-Cost-DS in minor-closed families of graphs

In this section, we give a primal-dual algorithm for the minimum cost Dominating Set problem (Min-Cost-DS) in minor-closed families of graphs that achieves a constant factor approximation. The algorithm will also establish a matching upper bound on the integrality gap of the standard LP relaxation for the problem that was given in Section 2.

Let $G = (V, E)$ be a node-weighted graph, and let $\mathrm{cost}(v)$ denote the cost of v. As before, for each vertex v, we let $\Gamma(v)$ denote the set of all neighbors of v in G. Let $\Gamma^+(v) = \Gamma(v) \cup \{v\}$. The primal and dual LPs are described below; we omit the constraint $x(v) \leq 1$ from the primal LP, since it is redundant.

Min-Cost-DS-LP			Dual of Min-Cost-DS-LP		
min	$\sum_{v \in V} x(v)\mathrm{cost}(v)$		max	$\sum_{v \in V} y(v)$	
s.t.	$\sum_{u \in \Gamma^+(v)} x(u) \geq 1$	$v \in V$	s.t.	$\sum_{u \in \Gamma^+(v)} y(u) \leq \mathrm{cost}(v)$	$v \in V$
	$x(v) \geq 0$	$v \in V$		$y(v) \geq 0$	$v \in V$

The algorithm is based on the primal-dual framework of Goemans and Williamson [10]. The algorithm selects a dominating set X for G. Initially, X consists of all vertices with zero cost. We also maintain a dual solution $\mathbf{y}$; initially, $y(v) = 0$ for all $v \in V$. We proceed in iterations. Consider an iteration i and let X_{i-1} be the set of nodes selected in the first $i-1$ iterations. Let A_i be the set of all vertices $v \in V$ such that $X_{i-1} \cap \Gamma^+(v)$ is empty. If A_i is empty, X_{i-1} is a dominating set and we return X_{i-1}. Otherwise, we increase the dual variables $\{y(a) \mid a \in A_i\}$ uniformly until a dual constraint for a node v becomes tight, i.e., we have $\sum_{u \in \Gamma^+(v)} y(v) = \mathrm{cost}(v)$; we add all the tight vertices to X.

We note that, in each iteration i, it is possible to increase the dual variables corresponding to the nodes of A_i. The set X_{i-1} contains all the vertices whose dual constraints are tight at the beginning of iteration i. Thus, at the beginning of iteration i, for each vertex $a \in A_i$ and each vertex v such that $a \in \Gamma^+(v)$, the dual constraint corresponding to v is slack, i.e., we have $\sum_{u \in \Gamma^+(v)} y(u) < \mathrm{cost}(v)$. Therefore the algorithm terminates in at most n iterations.

Finally, we perform a *reverse-delete step*. Let X be the dominating set selected by the primal-dual algorithm. We select a subset Y as follows. We start with $Y = X$. We order the vertices of Y in the reverse of the order in which they were selected by the primal-dual algorithm. We consider the

6

vertices of Y in this order. Let v be the current vertex. If $Y - v$ is a dominating set, we remove v from Y.

The algorithm described above is well-defined on general graphs, but its approximation is $\Omega(n)$. In the following, we show that we can take advantage of the fact that minor-free graphs are sparse in order to show that the algorithm achieves a constant factor approximation in minor-closed families of graphs; the constant depends on the family.

We start by noting that the dual solution $\mathbf{y}$ satisfies the complementary slackness conditions.

Proposition 3.1. *For each vertex $v \in Y$, we have $\sum_{u \in \Gamma^+(v)} y(u) = \mathrm{cost}(v)$.*

The following lemma gives us a very convenient way to upper bound the approximation ratio. The lemma follows from a standard primal-dual analysis and the fact that the algorithm increases the dual variables uniformly in each iteration. Recall that Y is the final dominating set after performing reverse-delete, and X_{i-1} is the set of vertices selected in the first $i - 1$ iterations of the algorithm.

Lemma 3.2. *Let $W_i = Y - X_{i-1}$. Suppose that there exists a γ such that, for each iteration i of the algorithm, we have*
$$\sum_{v \in A_i} \left| W_i \cap \Gamma^+(v) \right| \le \gamma \left| A_i \right|.$$

Then the cost of Y is at most $\gamma \cdot \mathrm{OPT}$, where OPT is the cost of the optimal solution to Min-Cost-DS-LP.

Proof: By Proposition 3.1, we have
$$\sum_{v \in Y} \mathrm{cost}(v) = \sum_{v \in Y} \sum_{u \in \Gamma^+(v)} y(u).$$

By rearranging the second summation, we get that
$$\sum_{v \in Y} \mathrm{cost}(v) = \sum_{v \in Y} \sum_{u \in \Gamma^+(v)} y(u) = \sum_{v \in V} y(v) \left| Y \cap \Gamma^+(v) \right|.$$

Since $\mathbf{y}$ is a feasible dual solution, by weak duality, we have
$$\mathrm{OPT} \ge \sum_{v \in V} y(v).$$

Therefore it suffices to show that
$$\sum_{v \in V} y(v) \left| Y \cap \Gamma^+(v) \right| \le \gamma \sum_{v \in V} y(v).$$

We can prove the inequality above by induction on the number of iterations. Initially, $y(v) = 0$ for all vertices v and the inequality clearly holds. Now consider an iteration $i \ge 1$. Let ϵ be the amount by which the dual variables $\{y(a) \mid a \in A_i\}$ are increased in iteration i. The right-hand side of the inequality increases by $\epsilon |A_i|$. Thus, if we can show that the left-hand side increases by at most $\epsilon \gamma |A_i|$, the inequality will follow. The left hand side of the inequality increases by $\epsilon \sum_{v \in A_i} |Y \cap \Gamma^+(v)|$. For each $v \in A_i$, we have $\Gamma^+(v) \cap X_{i-1}$ is empty, and thus
$$\sum_{v \in A_i} \left| Y \cap \Gamma^+(v) \right| = \sum_{v \in A_i} \left| W_i \cap \Gamma^+(v) \right| \le \gamma \left| A_i \right|,$$

7

where the last inequality follows from the assumption in the statement of the lemma. Therefore the left-hand side increases by at most $\epsilon\gamma\,|A_i|$, and the lemma follows. $\qquad\square$

Therefore, in order to upper bound the approximation ratio of the algorithm, it suffices to prove the following key lemma. The lemma follows from the minimality of Y and the fact that minor-free graphs are sparse, in the sense that the number of edges is proportional to the number of vertices.

Lemma 3.3. *Suppose that the input graph G belongs to a minor-closed familty $\mathcal{G}$ of graphs. There is a constant $c(\mathcal{G})$ depending only on $\mathcal{G}$ such that, for each iteration i of the algorithm, we have*

$$\sum_{u\in A_i}\left|W_i\cap\Gamma^+(u)\right|\le c(\mathcal{G})\cdot|A_i|\,,$$

where $W_i = Y - X_{i-1}$.

We devote the rest of this section to the proof of Lemma 3.3. We will prove the lemma in two steps. In the first step, we use the sparsity of minor-free graphs to show that the sum $\sum_{u\in A_i}|W_i\cap\Gamma^+(u)|$ is at most a constant times larger than $|A_i| + |W_i|$. In the second step, we use the minimality of W_i to show that $|W_i| \le |A_i|$.

Lemma 3.4. *Let $c'(\mathcal{G})$ be a constant such that, for each graph $K \in \mathcal{G}$, we have $|E(K)| \le c'(\mathcal{G})\,|V(K)|$. For each iteration i, we have*

$$\sum_{u\in A_i}\left|W_i\cap\Gamma^+(u)\right|\le|A_i\cap W_i|+c'(\mathcal{G})(|A_i|+3\,|W_i|).$$

Proof: Consider an iteration i of the algorithm. We have

$$\sum_{u\in A_i}\left|W_i\cap\Gamma^+(u)\right|=|A_i\cap W_i|+\sum_{u\in A_i}|W_i\cap\Gamma(u)|\,.$$

We can upper bound the second sum in the equation above as follows. Let G_1 be the subgraph of G whose vertices are $A_i\cup W_i$ and whose edges are all the edges of G with one endpoint in $A_i - W_i$ and the other in W_i. Note that $\sum_{u\in A_i-W_i}|W_i\cap\Gamma(u)|$ is equal to the number of edges of G_1. Let G_2 be the subgraph of G whose vertices are W_i and whose edges are all the edges of G with one endpoint in $A_i\cap W_i$ and the other in $W_i - A_i$. Finally, let $G_3 = G[A_i\cap W_i]$ be the subgraph of G induced by $A_i\cap W_i$. Note that $\sum_{u\in A_i\cap W_i}|W_i\cap\Gamma(u)|$ is equal to the number of edges of G_2 plus the number of edges of G_3. Therefore we have

$$\sum_{u\in A_i}|W_i\cap\Gamma(u)|=|E(G_1)|+2\,|E(G_2)|+|E(G_3)|\,.$$

Therefore we have

$$\sum_{u\in A_i}|W_i\cap\Gamma(u)|\le c'(\mathcal{G})(|V(G_1)|+2\,|V(G_2)|+|V(G_3)|)=c'(\mathcal{G})(|A_i|+3\,|W_i|).$$

$\qquad\square$

Lemma 3.5. *For each iteration i, we have $|W_i| \le |A_i|$.*

Proof: Consider a vertex $w \in W_i$. We claim that, since we could not remove w in the reverse-delete step, there is a vertex $v \in A_i$ such that $\Gamma^+(v)\cap(X_{i-1}\cup Y) = \{w\}$. We can show this as follows. Since we could not remove w, there is a vertex $v \in V - (Y\cup X_{i-1})$ such that $\Gamma^+(v)\cap(Y\cup X_{i-1}) = \{w\}$.

Since v is not dominated by X_{i-1}, v is in A_i. Thus each vertex $w \in W_i$ has a *witness* vertex $v \in A_i$ such that $\Gamma^+(v) \cap (X_{i-1} \cup Y) = \{w\}$. Now we claim that each vertex $v \in A_i$ is a witness vertex for at most one vertex of W_i. Suppose for contradiction that a vertex $v \in A_i$ is a witness vertex for two vertices w_1 and w_2 in W_i. Without loss of generality, w_1 was selected by the algorithm after w_2. Consider the iteration of the reverse-delete step that considered w_1. At this point w_2 had not been considered yet and thus it is in Y. Thus $w_2 \in \Gamma^+(v) \cap (X_{i-1} \cup Y)$, which contradicts the fact that $\Gamma^+(v) \cap (X_{i-1} \cup Y) = \{w_1\}$. Therefore $|W_i| \leq |A_i|$, as desired. $\square$

Lemma 3.3 follows from Lemma 3.4 and Lemma 3.5. We have the following upper bounds on the constant $c'(\mathcal{G})$ (see Lemma 3.4). If $\mathcal{G}$ is a minor-closed family, there is a constant-sized graph H such that $\mathcal{G}$ is the family of all graphs that do not have H as a minor. As shown by Kostochka [20], we have $c'(\mathcal{G}) = O(\sqrt{\log(|V(H)|)})$ for the family of H-minor-free graphs. If G is a planar graph, we have $c'(\mathcal{G}) < 3$; if G is also bipartite, the constant improves to 2. Thus the algorithm achieves an 10-approximation for planar graphs.

Remark 3.6. *The algorithm above can be easily adapted to give a constant factor approximation for the minimum cost* Steiner Dominating Set *problem in minor-closed families of graphs. In the Steiner problem, we are given a subset of vertices called terminals and the goal is to select a set that dominates the terminals.*

Remark 3.7. *A constant factor approximation for the minimum cost* Steiner Dominating Set *problem in minor-free graphs can also be obtained via iterated rounding.*

References

[1] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.

[2] A. A. Benczúr and D. R. Karger. Approximating st minimum cuts in $\tilde{O}(n^2)$ time. In *Proc. of ACM STOC*, pages 47–55, 1996.

[3] G. Călinescu, C. Chekuri, and J. Vondrák. Disjoint bases in a polymatroid. *Random Structures & Algorithms*, 35(4):418–430, 2009.

[4] R. Carr and S. Vempala. Randomized metarounding. *Random Structures & Algorithms*, 20(3):343–352, 2002.

[5] K. Censor-Hillel, M. Ghaffari, and F. Kuhn. A new perspective on vertex connectivity. *arXiv preprint arXiv:1304.4553*, 2013.

[6] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proc. of International Conference on Communications*, volume 1, pages 376–380, 1997.

[7] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad hoc networks using a spine. In *Proc. of International Conference on Computer Communications and Networks*, pages 34–39, 1997.

[8] E. D. Demaine, M. Hajiaghayi, and P. N. Klein. Node-weighted Steiner tree and group Steiner tree in planar graphs. In *Proc. of ICALP*, pages 328–340. Springer, 2009.

[9] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM Journal on Computing*, 32(1):172–195, 2002.

[10] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296, 1995.

[11] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.

[12] S. Guha, A. Moss, J. S. Naor, and B. Schieber. Efficient recovery from power outage. In *Proc. of ACM STOC*, pages 574–582, 1999.

[13] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Domination in graphs: advanced topics*, volume 40. Marcel Dekker, 1998.

[14] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of domination in graphs*. Marcel Dekker, 1998.

[15] S. T. Hedetniemi and R. C. Laskar. *Topics on domination*, volume 48. North Holland, 1991.

[16] D. R. Karger. Using randomized sparsification to approximate minimum cuts. In *Proc. of ACM-SIAM SODA*, pages 424–432, 1994.

[17] D. R. Karger. A randomized fully polynomial time approximation scheme for the all terminal network reliability problem. In *Proc. of ACM STOC*, pages 11–17, 1995.

[18] D. R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999. Preliminary version in STOC 1994.

[19] D. R. Karger and M. S. Levine. Finding maximum flows in undirected graphs seems easier than bipartite matching. In *Proc. of ACM STOC*, pages 69–78, 1998.

[20] A. V. Kostochka. Lower bound of the Hadwiger number of graphs by their average degree. *Combinatorica*, 4(4):307–316, 1984.

[21] D. Mahjoub and D. W. Matula. Employing (1- ε) dominating set partitions as backbones in wireless sensor networks. In *Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 98–111, 2010.

[22] R. Misra and C. Mandal. Rotation of cds via connected domatic partition in ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 8(4):488–499, 2009.

[23] T. Moscibroda and R. Wattenhofer. Maximizing the lifetime of dominating sets. In *Proc. of International Parallel and Distributed Processing Symposium*, 2005.

[24] S. V. Pemmaraju and I. A. Pirwani. Energy conservation via domatic partitions. In *Proc. of International symposium on Mobile ad hoc networking and computing*, pages 143–154, 2006.