Distributed algorithms for weighted problems in minor-closed families *

A. Czygrinow
Department of Math and Stat
Arizona State University
Email:andrzej@math.la.asu.edu.

M. Hańćkowiak
Faculty of Math and CS
Adam Mickiewicz University
Email: mhanckow@main.amu.edu.pl.

Abstract

We give efficient distributed algorithms for weighted versions of the maximum matching problem and the minimum dominating set problem for graphs from minor-closed families. To complement these results we argue that no efficient distributed algorithm for the minimum weight connected dominating set exists.

Keywords: Distributed algorithms, minor-closed families of graphs, dominating set.

*This work was supported by grant N206 017 32/2452 for years 2007-2010.

1 Introduction

Efficient distributed algorithms for only very few graph-theoretic problems are known. At the same time there has been much more success in designing efficient distributed algorithms in case the underlying topology of the network has additional properties. For example, many problems can be solved efficiently in constant degree graphs and some problems admit rather easy distributed algorithms in graphs of bounded arboricity (for example in planar graphs). In this paper, we will study distributed complexity of three fundamental problems in proper minor-closed families of graphs. We will show that the maximum-weight matching problem and the minimum-weight dominating set problem admit efficient distributed approximations but the minimum-weight connected dominating set problem does not. This extends and complements the results from [CH06c] where distributed complexity of unweighted versions of the above problems is analyzed. Note however that algorithms for weighted problems are significantly different than the ones from [CH06c]. In fact, even the distributed complexity of weighted and unweighted problems can be different. For example, in [CH06c] we proved that the minimum-connected dominating set problems admits an efficient distributed approximation in connected graphs which come from minor-closed families. This is not the case for the weighted analog as we argue in the last section of this paper. The algorithms for weighted versions of the maximum matching and the minimum dominating set problems are in turn based on a completely new and provably more powerful partitioning algorithm than the corresponding clustering procedure used before.

1.1 Terminology and notation

We will consider the message-passing distributed model (see Linial [L92]). In this model, network is represented by an undirected graph with vertices corresponding to processors, and edges corresponding to communication links between processors. The network is synchronized and computations proceed in discrete rounds. In a single round a vertex can send and receive messages from its neighbors, and in addition can perform some local computations. Neither the amount of local computations nor the lengths of messages are restricted in any way. We will also assume that nodes in the network have unique identifiers which are positive integers from $\{1, \ldots, |G|\}$ where $|G|$ is the order of G and is globally known.

Although different possible measures of efficiency of a distributed algorithm can be assumed, following [L92] we call a distributed algorithm *efficient* if it runs in a poly-logarithmic (in the order of the graph) number of rounds. Consequently, if the diameter of the underlying network is poly-logarithmic then any problem admits a trivial efficient solution. In this paper, we shall focus on distributed approximation algorithms for minor-closed families. All graphs are simple and in the graph-theoretic terminology we will follow [D97]. A graph H is called a *minor* of G if it can be obtained from a subgraph of G by a series of edge contractions. A family $\mathcal{C}$ is called *minor-closed* if for any graph $G \in \mathcal{C}$ every minor of G is also in $\mathcal{C}$. A family $\mathcal{C}$ is proper if there is a graph G which is not in $\mathcal{C}$ and is non-trivial if it contains a graph with at least one edge. We will always assume that our minor-closed family is both proper and non-trivial. Clearly if a minor-closed family is not proper then it is simply the family of all graphs and our algorithms do not work, if it is trivial then it contains only graphs which are unions of isolated vertices and the problems which we want to solve are trivial. The most important example of a minor-closed family is the class of all planar graphs. Recently algorithmic questions for different minor-closed classes of graphs like for example the family of graphs with a bounded tree-width or a bounded genus have attracted attention. Let $\mathcal{C}$ be a minor-closed family and let $\rho = \sup_{G \in \mathcal{C}} \frac{||G||}{|G|}$ be the edge density of $\mathcal{C}$. It is known (see for example [NM05]) that ρ is finite if and only if $\mathcal{C}$ is a proper minor-closed family. We will write $\mathcal{C}_\rho$ for a minor-closed family with edge density ρ and assume that ρ is known by an algorithm.

A *matching* in graph G is a subset M of edges of G with no two edges from M sharing a vertex. For an edge-weighted graph $(G, \bar{\omega})$ with $\bar{\omega} : E(G) \to R^+ \cup \{0\}$ we denote by $\beta(G)$ the maximum weight of a matching in G, that is $\beta(G) = \max_M \sum_{e \in M} \bar{\omega}(e)$. A *dominating set* in a graph G is a subset D of vertices such that for every vertex $v \notin D$ a neighbor of v is in D. For a vertex-weighted graph (G, ω)

with $\omega : V(G) \to R^+ \cup \{0\}$ we denote by $\gamma(G)$ the minimum weight of a dominating set in G, that is $\gamma(G) = \min_D \sum_{v \in D} \omega(v)$. Finally, a dominating set D is called a *connected dominating set* in G if it is a dominating set and the subgraph of G induced by D is connected. We will denote by $\gamma_c(G)$ the minimum weight of a connected dominating set in a connected graph G.

We will denote by $|G|$ the order of G, that is the number of vertices of G and by $||G||$ the size of G, that is the number of edges of G. As already noted we will assume that each vertex v has a unique identifier $ID(v)$ and $ID : V(G) \to \{1, \ldots, |G|\}$. Since our partitioning algorithm will be applied to an auxiliary graph of G and it will be important to distinguish between the range of identifiers in the auxiliary graph and the order of the graph we denote by $ID(H) = \bigcup_{v \in V(H)} \{ID(v)\}$.

1.2 Results

We will give distributed approximation algorithms for maximum-weight matching problem and for the minimum-weight dominating set problem for graphs from a minor-closed family $\mathcal{C}_\rho$. In the case of the maximum-weight matching problem we will give a distributed algorithm which given a positive integer d finds in an edge-weighted graph $(G, \bar{\omega})$ with $G \in \mathcal{C}_\rho$ a matching M of weight $\bar{\omega}(M) \geq \left(1 - \frac{1}{\log^d |G|}\right) \beta(G)$. The algorithm runs in a poly-logarithmic number of rounds. (Theorem 3.1.) For the minimum-weight dominating set problem, we will prove that there is a distributed algorithm which given a positive integer d finds in vertex-weighted graph (G, ω) with $G \in \mathcal{C}_\rho$ a dominating set D such that $\omega(D) \leq \left(1 + \frac{1}{\log^d |G|}\right) \gamma(G)$. This algorithm again runs in a poly-logarithmic number of rounds. (Theorem 3.4.) For the minimum-weight connected dominating set problem we will argue that to accomplish any finite multiplicative approximation error, $\Omega(|G|)$ rounds are needed. Both algorithms use a vertex partitioning procedure which partitions the vertex set of a graph G into sets $V_1, \ldots, V_l$ so that each $G[V_i]$ has a poly-logarithmic diameter and the weight of the border vertices is small with respect to the total weight of G (see Corollary 2.10 for a precise statement).

1.3 Related Work

We will briefly put our results in a more general context. The reader is directed to Elkin's survey [E04] for a more comprehensive overview of distributed approximation algorithms. Let us first note that efficient distributed algorithms that find exact solutions to the above problems do not exist (even for unweighted analogs). For example, the minimum dominating set problem and the maximum matching problem when restricted to a cycle G cannot be found in $o(|G|)$ rounds ([L92]). In addition, to achieve a poly-logarithmic approximation ratio for minimum dominating set at least $\max\{\Omega(\sqrt{\log |G| / \log \log |G|})\Omega(\log \Delta / \log \log \Delta)\}$ rounds are required ([KMW04]).

Distributed approximation algorithms for planar graphs were studied in [CH06a] and [CHS06]. In particular, [CH06a] contains an efficient distributed approximation for the maximum-weight independent set problem in planar graphs. For the unweighted problems, [CH06c] contains efficient distributed algorithms for unweighted versions of the three problems considered in this paper. There has also been some success in designing efficient distributed approximations for other than minor-closed families of graphs. As mentioned before, if a graph has a constant maximum degree then many problems can be solved efficiently. In addition, [KMNW05] and [CH06b] contain approximations for unweighted versions of the maximum independent set , maximum matching, minimum dominating set, and minimum connected dominating set problems in unit-disk graphs.

1.4 Organization

In the rest of the paper we will first discuss vertex partitioning problems in weighted graphs and give our main auxiliary procedure (Section 2). Then, in Section 3, we give approximation algorithms and discuss the minimum-weight connected dominating set problem.

2 Partitioning of vertex-weighted graphs

We will start with fixing some general graph-theoretic terminology. For a graph G, $V(G)$ will denote the vertex set of G and $E(G)$ will denote the edge set of G. If U, U' are two disjoint subsets of $V(G)$ then $E_G(U, U')$ denotes the set of edges with one endpoint in U, another in U'. For $v \in V(G)$, $N(v)$ denotes the set of neighbors of v in G and if $U \subseteq V(G)$ then $N_G(U) := \bigcup_{u \in U} N(u) \setminus U$. For two vertices u, u', $dist_G(u, u')$ is the length of the shortest path between u and u', the diameter of G, $diam_G$, is the maximum of $dist_G(u, u')$ over all pair (u, u'), and for sets U, U', we set $dist_G(U, U') := \min_{u \in U, u' \in U'} dist_G(u, u')$. In addition for a subgraph H of G we will consider two different diameters of H. The strong diameter of H, $SDiam_G(H)$, will be defined as as $diam_H$ and the weak diameter of H, $WDiam_G(H)$, will be defined as $max_{u,u \in V(H)} dist_G(u, u')$. Clearly $WDiam_G(H) \leq SDiam_G(H)$

Let $\mathcal{C}_\rho$ be a minor-closed family of graphs G with the edge-density ρ, that is

$$\rho = \sup \frac{||G||}{|G|}$$

where the supremum is taken over all graphs from $\mathcal{C}_\rho$. It is known (see [NM05] for this and many other results) that ρ is finite if and only if $\mathcal{C}_\rho$ is a proper minor-closed family. In addition, in the paper, we will always assume that $\mathcal{C}_\rho$ is proper and $\rho > 0$.

Although vast majority of the paper is concerned with vertex-weighted graphs, we will start with a brief discussion that shows a connection between distributed partitioning problems for vertex-weighted and edge-weighted graphs (Section 2.1). In Section 2.2, we will give an efficient distributed partitioning algorithm for vertex-weighted graphs from $\mathcal{C}_\rho$. The distributed algorithm is deterministic but we assume that both $|G|$ and ρ are known to all vertices of G.

2.1 Weighted graphs

For a graph $G \in \mathcal{C}_\rho$ we will consider two types of weight functions on G. Pair (G, ω) with $\omega : V(G) \to R^+ \cup \{0\}$ will be called *vertex-weighted graph* G and the pair $(G, \bar{\omega})$ with $\bar{\omega} : E(G) \to R^+ \cup \{0\}$ will be called *edge-weighted graph* G. We need some more notation and terminology. Let (G, ω) be a vertex-weighted graph. For a set $S \subseteq V(G)$ we define $\omega(S) := \sum_{v \in S} \omega(v)$. A vertex of S is called a *border vertex* in S if it has a neighbor in $V(G) \setminus S$. The set of all border vertices in S is denoted by $\partial(S)$ and for a partition $P = (V_1, V_2, \dots, V_l)$ of $V(G)$ we set $\partial(P) := \bigcup_{i=1}^{l} \partial(V_i)$. In the case of an edge-weighted graph $(G, \bar{\omega})$ we define $\bar{\partial}(S)$ to be the set of all edges with one endpoint in S and another in $V(G) \setminus S$. Then for a partition $P = (V_1, V_2, \dots, V_l)$ of $V(G)$, $\bar{\partial}(P) := \bigcup_{i=1}^{l} \bar{\partial}(V_i)$.

Definition 2.1 *Let (G, ω) be a vertex-weighted graph and let $a(\cdot), b(\cdot)$ be functions to R. A partition $P = (V_1, V_2, \dots, V_l)$ of $V(G)$ is called an (a, b)-vertex-weight partition if the following two conditions are satisfied:*

- *For $i = 1, \dots, l$, $G[V_i]$ is connected and $WDiam_G(G[V_i]) \leq a(|G|)$.*
- *$\omega(\partial(P)) \leq \omega(V(G))/b(|G|)$.*

Similarly we define an (a, b)-edge-weight partition of $(G, \bar{\omega})$. We will be almost exclusively interested in cases when both a and b are poly-logarithmic functions. In [CH06a], a distributed algorithm that finds a $(\log |G|, \log |G|)$- edge-weight partition of an edge-weighted planar graph G is given. The algorithm runs in a poly-logarithmic number of rounds. This edge-weight partition can be used to give distributed approximation algorithms for the maximum-weight independent set problem. In addition, a similar procedure can be used to give distributed approximations for the unweighted versions of the maximum matching problem or the minimum dominating set problem in graphs G which come from a fixed minor-closed family. On the other hand, the edge-weight partition property is not strong enough to yield approximations for weighted analogs of the maximum matching problem or the minimum dominating set problem. As we will show in the next section, vertex-weight partition can be found by a distributed algorithm efficiently and can be used to design approximations for the weighted versions of the above two problems. Let us first note however the vertex-weight partition is indeed stronger than an edge-weight partition.

Fact 2.2 *Let G be a graph. For every $\bar{\omega} : E(G) \to R^+ \cup \{0\}$ there exists $\omega : V(G) \to R^+ \cup \{0\}$ such that if $P = (V_1, \ldots, V_l)$ is an (a, b)-vertex-weight partition of (G, ω) then P is the (a', b')-edge-weight partition of $(G, \bar{\omega})$ with $a' = a$ and $b' = b/2$.*

Proof. Set $\omega(u) := \sum_{v \in N(u)} \bar{\omega}(u, v)$ and note that $\sum_v \omega(v) = 2 \sum_e \bar{\omega}(e)$. Suppose that $P = (V_1, \ldots, V_l)$ is an (a, b)-vertex-weight partition of (G, ω). Then $\omega(\partial(P)) \le \omega(V(G))/b(|G|)$ and clearly the total weight of edges incident to $\partial(P)$ is at most $\omega(\partial(P))$. Consequently,

$$\bar{\omega}(\bar{\partial}(P)) \le \omega(V(G))/b(|G|) \le 2 \cdot \bar{\omega}(E(G))/b(|G|).$$

□

2.2 Partitioning Algorithm

We will now give an algorithm which finds an (a, b)-vertex-weight partition. Let us start by fixing some additional terminology. Let G be a graph from $\mathcal{C}_\rho$ and let $\omega : V(G) \to R^+ \cup \{0\}$. A small modification (change in the number of iterations) of the algorithms Clustering and WISPlanar from [CH06a] yields the following two facts.

Lemma 2.3 *Let $\mathcal{C}_\rho$ be a minor-closed family of graphs. Let $G \in \mathcal{C}_\rho$ and let $(G, \bar{\omega})$ be an edge-weighted graph. There exists a distributed algorithm which given a constant $d > 1$ finds in $O(\log |G| \log^* |G|)$ rounds a (D_ρ, d)-edge-weight partition for some constant $D_\rho = D_\rho(d)$.*

Lemma 2.4 *Let $\mathcal{C}_\rho$ be a minor-closed family of graphs. Let $G \in \mathcal{C}_\rho$ and let (G, ω) be a vertex-weighted graph. There exists a distributed algorithm which given a constant $d > 2\rho$ finds in $O(\log |G| \log^* |G|)$ rounds a maximal independent set I in G with*

$$\omega(I) \ge \omega(V(G))/d.$$

Our first procedure, Heavy Subset, finds a subset of vertices of a large weight which induces subgraphs of small weak diameter. As the procedure is a bit technical we will divide it into two phases.
Heavy Subset Phase 1. Use the algorithm Decomposition from [CH06c] to find a partition $(V_1, \ldots, V_k)$ of $V(G)$ with properties that each V_i is an independent set, $k = O(\log |G|)$, and for every i, if $v \in V_i$ then $|N(v) \cap \bigcup_{j>i} V_j| \le 3\rho$. Give the orientation (u, v) (from u to v) to every edge $\{u, v\}$ with $u \in V_i$ and $v \in V_j$ whenever $i < j$ and define the weight of (u, v) by setting $\bar{\omega}(u, v) := \omega(u)$. Note that the out-degree of this directed graph is at most 3ρ. Let $d := 3 \cdot \rho / \left(1 - \frac{1}{2\rho+1}\right)$ and let D_ρ denote the constant from Lemma 2.3. Find a (D_ρ, d)-edge-weight partition $(V_1, \ldots, V_k)$ of $(G, \bar{\omega})$. Consider two sets of vertices: B (black) and W (white). Set initially $B := V(G)$ and $W := \emptyset$. For every vertex u, in parallel, if $u \in V_i$ and there is a vertex $v \in V \setminus V_i$ such that (u, v) is an arc, then change the color of u to white. We will end the phase one here. First note the following fact.

Fact 2.5 *After* Heavy Subset Phase 1 *all edges with endpoints in different V_i's have at least one endpoint in W.*

In addition, we have the following lemma.

Lemma 2.6 *Let B be the set of black vertices in G after* Heavy Subset Phase 1. *We have*

$$\omega(B) \ge \frac{\omega(V(G))}{2\rho + 1}.$$

Proof. From Lemma 2.3 the weight of edges going across partition classes $V_1, \ldots, V_k$ is less than or equal to $\left(\left(1 - \frac{1}{2\rho+1}\right)/(3 \cdot \rho)\right) \bar{\omega}(E(G))$ which is at most $\left(1 - \frac{1}{2\rho+1}\right) \omega(V(G))$. For every vertex u if $u \in V_i$ and u has an out-neighbor v in $V \setminus V_i$ then $\omega(u) = \bar{\omega}(u, v)$ and so $\omega(W) \leq \left(1 - \frac{1}{2\rho+1}\right) \omega(V(G))$. □

HEAVY SUBSET PHASE 2. After the execution of phase one $(V_1, \ldots, V_k)$ is a partition of $V(G)$ every edge with endpoints in different V_i's has at least one endpoint in W. Consequently some of the border vertices of each V_i can be white. A vertex w is called a *troubler* if $w \in W$ and for some $v_i \in V_i \cap B$ and $v_j \in V_j \cap B$ with $i \neq j$, $v_i w v_j$ is a path (of length two) in G. In other word a troubler is a white vertex which is connected by an edge with two black vertices in different V_i's. Clearly only a border vertex can be a troubler. Recall that in phase one we gave an orientation to all edges of G. We shall now define an auxiliary hypergraph. For each troubler w, if w is in V_i and has more than one out-neighbor in $B \cap (V \setminus V_i)$ then consider the hyper-edge f_w consisting of these out-neighbors and let H be the hypergraph $\mathcal{H} := (B, \bigcup\{f_w\})$. Note that as $\mathcal{H}$ is on B, there can be many isolated vertices in $\mathcal{H}$. In addition $|f_w| \leq 3\rho$ for any troubler w as the out-degree is at most 3ρ.

Next task is to find a "heavy" maximal independent set I in $\mathcal{H}$. This is done by consider the graph G' with $V(G') := B$ and the edge set $E(G')$ obtained in the following way. Every troubler w selects two distinct vertices $u, v \in f_w$ and adds the edge $\{u, v\}$ to $E(G')$. Then every edge in G' corresponds to a path uwv in G with $w \in W$ and different paths contain different w's. Therefore G' is a topological minor of G and so $G' \in \mathcal{C}_\rho$. Use Lemma 2.4 to find a maximal independent set I in G' with $\omega(I) \geq \omega(B)/(2\rho + 1) \geq \omega(V(G))/(2\rho + 1)^2$ and repaint vertices from $B \setminus I$ with the white color. Repeat the process after updating sets f_w and the hypergraph $\mathcal{H}$. Note that in each round of the above procedure, the size of f_w drops by at least one and so after $3\rho - 1$ rounds $|f_w| \leq 1$ for every troubler w. Consequently, the last instance of I from the loop above is an independent set in the initial hypergraph $\mathcal{H}$. In addition we see that I has a large weight.

Fact 2.7 *Set I of vertices in G is an independent set in the hypergraph $\mathcal{H}$ and $\omega(I) \geq \omega(V(G))/(2\rho + 1)^{3\rho}$.*

Now for every troubler w with $|f_w| = 1$ let u_w denote the vertex in f_w and let $S = \{\{w, u_w\} : |f_w| = 1\}$. Consider the subgraph $\tilde{G}_i$ of $G[V_i]$ induced by edges which have at least one endpoint in $B \cap V_i$. In addition, consider another auxiliary graph G'': For every $i = 1, \ldots, k$, contract every component U of the subgraph $\tilde{G}_i$ to a vertex and let v_U denote the vertex obtained from set U. Put an edge between two vertices $v_U, v_{U'}$ whenever $E_G(U, U') \cap S \neq \emptyset$. In addition, set $\omega(v_U) := \sum_{w \in B \cap U} \omega(w)$. Finally, note that the graph G'' is in $\mathcal{C}_\rho$ and so by Lemma 2.4, we can find an independent set I in G'' of weight which is at least $\omega(V(G''))/(2\rho + 1)$ which by Fact 2.7 is at least $\omega(V(G))/(2\rho + 1)^{3\rho+1}$. Now repaint all vertices of B which are not in a set U with $v_U \in I$ with the white color.

Finally consider the subgraph $\tilde{G}$ of G induced by edges from $E(G)$ which have at least one endpoint in B and return the components of $\tilde{G}$. This is the end of phase two.

Lemma 2.8 *Let B be the set of black vertices obtained in Phase 2 and let $L_1, L_2, \ldots, L_p$ be the components of $\tilde{G}$ which are returned in the end of phase two. We have*

- $\omega(B) \geq \omega(V(G))/(2\rho + 1)^{3\rho+1}$.
- *For $i = 1, \ldots, p$, $WDiam_G(L_i) \leq D_\rho + 2$.*

Proof. We have already proved the first part. Recall that I denotes the independent set in G'' obtained in HEAVYSUBSET PHASE 2 and let $v_U, v_{U'} \in I$. We will first show that $dist_G(U \cap B, U' \cap B) \geq 3$. To that end assume first that $U \subset V_i$ and $U' \subset V_j$ with $i \neq j$ and suppose that there is a path of length at most two with one endpoint in $U \cap B$ and another in $U' \cap B$. Clearly the path cannot have length one as every edge from $E_G(V_i, V_j)$ has one endpoint in W (Fact 2.5). Consequently the path has length two and has the form $v_i w v_j$ with $w \in W$. Thus w is a troubler after all of the iterations in $\mathcal{H}$ and either $v_i = u_w$ or $v_j = u_w$ which yields an edge between $v_U, v_{U'}$ in G'' and contradicts the fact that I is independent.

Now suppose that $U, U' \subset V_i$. Then the graphs induced by U, U' are components in $\tilde{G}_i$ and so the distance between $U \cap B$ and $U' \cap B$ in $G[V_i]$ is at least three. In addition, if there is vertex $w \in V_j$ for $j \neq i$ with a neighbor in $U \cap B$ and $U' \cap B$ then $w \in W$ and $|f_w| \geq 2$ which is not possible.

Now take a component L of $\tilde{G}$ and let $v_U \in I$ be such that L and U intersect in a black vertex. Also let i be such that $U \subseteq V_i$. If $N_{\tilde{G}}(U \cap B) \subseteq V_i$ then the vertex set of L is a subset of V_i and so the diameter of L is at most D_ρ. Otherwise take a vertex $u \in N_{\tilde{G}}(U \cap B) \setminus V_i$. Then u is white and so u is troubler. As there are no edges in $\tilde{G}$ with both endpoints white, u has at most two neighbors in $\tilde{G}$ and both of them are black. If one of them is in $U' \neq U$ then $dist_G(B \cap U, B \cap U') \leq 2$. Consequently u has one neighbor in $\tilde{G}$ from $B \cap U$. Therefore $V(L)$ is a subset of $N_G(U \cap B) \cup U \subseteq N_G(V_i) \cup V_i$. Since $WDiam_G(G[V_i]) \leq D_\rho$ we have $WDiam_G(L) \leq D_\rho + 2$. □

Now we can describe our main partitioning algorithm.

Vertex-weight Partition. Given is a vertex-weighted graph (G, ω) with $G \in \mathcal{C}_\rho$ and a positive integer t which can depend on $|G|$. Iterate with i from 1 to t. In the ith iteration, invoke Heavy Subset to find the set of black vertices B and the components $L_1, L_2, \ldots, L_p$ from Lemma 2.8. Obtain the minor G_i of G_{i-1} by contracting each of the $L_1, L_2, \ldots, L_p$ to a single vertex and set the weight of the vertex obtained from L_i to be equal to the total weight of white vertices in L_i. Let G_t be the graph obtained from G after all t iterations. For each vertex Q in G_t consider the set V_Q of all vertices in G which have been contracted to Q in the above iterations and return the partition $P = (V_Q | Q \in V(G_t))$.

Lemma 2.9 *Let $P = (V_1, V_2, \ldots, V_k)$ be the partition of graph $G \in \mathcal{C}_\rho$ obtained by* Vertex-weight Partition *with given parameter t.*

(a) Let D_ρ be the constant from Lemma 2.3 obtained by setting $d = 3 \cdot \rho / \left(1 - \frac{1}{2\rho+1}\right)$. Then

$$WDiam_G(G[V_i]) \leq (D_\rho + 3)^t.$$

(a) $\omega(\partial(P)) \leq \left(1 - \frac{1}{(2\rho+1)^{3\rho+1}}\right)^t \omega(V(G))$.

Proof. Let G_i denote the graph obtained after the ith iteration of Vertex-Weight Partition and let $G_0 := G$. To show (a), let $diam_i$ be the maximum of $WDiam_G(G[V'])$ over subsets $V' \subset V(G)$ that are contracted to single vertices in G_i. Clearly $diam_0 = 0$ and by Lemma 2.8 (part two) $diam_{i+1} \leq (D_\rho + 3) \cdot diam_i + D_\rho + 2$. Consequently

$$diam_t \leq (D_\rho + 3)^t.$$

To verify part (b), we consider the sequence of partitions $\{P_i\}$ where P_i is the partition of $V(G)$ obtained be creating a partition class for each vertex Q in G_i consisting of vertices that has been contracted to Q in iterations $1, \ldots, i$. Let $\partial_i = \partial(P_i)$ with $\partial_0 := V(G)$. Note that after the coloring of $V(G_i)$ performed by Heavy Subset only white vertices in a component L have neighbors in $V(G_i) \setminus L$. Therefore, $\omega(\partial_i)$ is smaller than or equal to the weight of white vertices in G_i. By definition of the weights in Vertex-Weight Partition, $\omega(V(G_{i+1}))$ is equal to the weight of white vertices in G_i and so $\omega(V(G_i)) \leq \left(1 - \frac{1}{(2\rho+1)^{3\rho+1}}\right)^{i-1} \omega(V(G))$ which in view of Lemma 2.8 (part one) gives

$$\omega(\partial_i) \leq \left(1 - \frac{1}{(2\rho+1)^{3\rho+1}}\right)^i \omega(V(G)).$$

□

For the next corollary, recall that $ID : V(G) \to \{1, \ldots, m\}$ is a function with $ID(v)$ equal to the identifier of vertex v. Although, as mentioned in the introduction, in the original graph G, $ID(V(G))$ is assumed to be equal to $V(G)$, in applications we will partition auxiliary graphs and it will be important to distinguish between $|G|$ and the order of the auxiliary graph.

Corollary 2.10 *(a) There is a distributed algorithm which given $0 < \epsilon < 1$ finds in a vertex-weighted graph (G, ω) with $G \in \mathcal{C}_\rho$ an (a, b)-vertex-weight partition $P = (V_1, \ldots, V_k)$ with $b \geq 1/\epsilon$ and $a \leq D(\epsilon)$ for some constant $D(\epsilon)$. The algorithm runs in $O(\log |G| \log^* |G|)$ rounds.*

(b) There is a distributed algorithm which given a positive integer p finds in a vertex-weighted graph (G, ω) with $G \in \mathcal{C}_\rho$ and $ID(v) \leq m$ for every $v \in V(G)$ an (a, b)-vertex-weight partition $P = (V_1, \ldots, V_k)$ with $b = \log^p m$ and $a = polylog(m)$. The algorithm runs in a pol-logarithmic (in m) number of rounds.

Proof. To obtain both algorithms we use Vertex-weight Partition. For the first one we have $t = \lfloor \log_u \epsilon \rfloor$ with $u = \left(1 - \frac{1}{(2\rho+1)^{3\rho+1}}\right)$. For the second one, we use $t = \lfloor -d \log_u \log m \rfloor$ in which case $(D_\rho + 3)^t$ is poly-logarithmic in m. Thus the weak diameter of each $G[V_i]$ is poly-logarithmic in m and each step executed in G_i (with $i \leq t$) can be done in a poly-logarithmic number of rounds in G. □

3 Applications

We will now show how to use the vertex-weight partition to design distributed approximations for the minimum-weight dominating set problem and the maximum-weight matching problem.

3.1 Matchings

Let us start with the maximum-weight matching problem. Let $(G, \bar{\omega})$ be an edge-weighted graph with $G \in \mathcal{C}_\rho$. In the algorithm, we first find a subgraph of G and use it to define the vertex-weighted graph (G, ω). Then we apply the partitioning procedure from Corollary 2.10. The procedure takes a positive integer d as an input.

ApproxMWM. Use the algorithm Decomposition from [CH06c] to find a partition of $V(G)$ into k sets $V_1, \ldots, V_k$ so that each V_i is an independent set, $k = O(\log |G|)$, and for every i, if $v \in V_i$ then $|N(v) \cap \bigcup_{j>i} V_j| \leq 3\rho$. For every vertex v if $v \in V_j$ then v properly colors all edges in $E(\{v\}, \bigcup_{j>i} V_j)$ using colors from $\{1, \ldots, 3\rho\}$. Let F_i be the subgraph of G induced by edges of color i. Then F_i is a forest every component of which has diameter $O(\log |G|)$. Find in each F_i a maximum weight matching N_i and let $Q := \bigcup N_i$. Now for every vertex v set $\omega(v) := \bar{\omega}(e)$ where e is an edge in Q of maximum weight which is incident to v. If no such edge exists set $\omega(v) := 0$. Use the algorithm from Corollary 2.10 (b) with $p = d+1$ and $m = |G|$ to obtain a vertex-weight partition $P = (V_1, \ldots, V_k)$ of (G, ω). Find a maximum weight matching M_i in each of $(G[V_i], \bar{\omega})$ and return $\bigcup M_i$.

Theorem 3.1 *Let $(G, \bar{\omega})$ be an edge-weighted graph with $G \in \mathcal{C}_\rho$. There is a distributed algorithm which given a positive integer d finds a matching M in G with*

$$\bar{\omega}(M) \geq \left(1 - \frac{1}{\log^d |G|}\right) \beta(G)$$

where $\beta(G)$ is the weight of a maximum-weight matching in G. The algorithm runs in a poly-logarithmic number of rounds.

Proof. We use ApproxMWM. Note that $\bar{\omega}(Q) \leq 3\rho\beta(G)$ and so the total vertex weight of G satisfies $\omega(V(G)) \leq 3\rho\beta(G)$. Moreover we have that, for every edge $\{u, v\} \in E(G)$, $\bar{\omega}(\{u, v\}) \leq \omega(u) + \omega(v)$. Indeed if $\{u, v\} \in Q$ then this is clear. If $\{u, v\}$ is not in Q and $\{u, v\}$ is in F_i then there exist at most two edges $e_1 = \{u, w\}, e_2 = \{v, z\}$ in M_i such that $\bar{\omega}(\{u, v\}) \leq \bar{\omega}(e_1) + \bar{\omega}(e_2)$. Consequently

$$\bar{\omega}(\{u, v\}) \leq \omega(u) + \omega(v).$$

We have

$$\omega(\partial(P)) \leq \omega(V(G))/\log^{d+1} |G| \leq \frac{3\rho\beta(G)}{\log^{d+1} |G|} \leq \frac{\beta(G)}{\log^d |G|}. \quad (1)$$

Every matching in G contains two types of edges: edges with both endpoints in some V_i and edges that are incident to $\partial(P)$. The total weight of the latter is at most $\frac{\beta(G)}{\log^d |G|}$ by (1) and so the matching M returned by ApproxMWM satisfies

$$\beta(G) \leq \omega(M) + \frac{\beta(G)}{\log^d |G|}.$$

□

3.2 Dominating sets

Let (G, ω) be a vertex-weighted graph. Recall that for any $D \subseteq V(G)$ we have $\omega(D) := \sum_{v \in D} \omega(v)$. We will denote by $\gamma(G) = \min \omega(D)$ where the minimum is taken over all dominating sets in graph G. For a vertex v, recall that $N(v)$ denotes the set of neighbors of v and $N[v] := N(v) \cup \{v\}$. Pick one vertex in $N[v]$, $s(v)$, with $\omega(s(v)) := \min_{w \in N[v]} \omega(w)$ and set $\bar{D} := \bigcup \{s(v)\}$.

Lemma 3.2 *Let (G, ω) be a vertex-weighted graph and let $\bar{D} := \bigcup \{s(v)\}$. Then $\bar{D}$ is a dominating set and $\omega(\bar{D}) \leq |G|\gamma(G)$.*

Proof. Clearly $\bar{D}$ is a dominating set. If D is a dominating set then for every v there a vertex $w_v \in D \cap N[v]$ and of course $\omega(w_v) \geq \omega(s(v))$. Consequently, $|G|\omega(D) \geq \sum_{v \in V} \omega(w_v) \geq \omega(\bar{D})$. □

Our approximation algorithm proceeds in two main phases. First we find a constant approximation of $\gamma(G)$ and next we find a more accurate approximation. We will repeatedly use the following type of a minor of G arising from a dominating set. For a dominating set D in G, every vertex $v \in V(G) \setminus D$ selects one vertex w in $N(v) \cap D$ and joints group U_w. Let G_D be obtained from G by contracting $U_w \cup \{w\}$ to a single vertex u_w with $\omega(u_w) := \omega(w)$. Then, clearly, $\omega(V(G_D)) = \omega(D)$.

Our algorithm ApproxMWDS is given a positive integer d which will be used in the second phase of the procedure.

ApproxMWDS Phase 1. Let $D := \bar{D}$. We iterate with i from 1 to $\log_2 |G|$. In the ith iteration, we consider (G_D, ω) and use Corollary 2.10 (a) with $\epsilon = 1/2$ to find a vertex-weight partition $(V_1', \ldots, V_k')$ of G_D. This gives a partition P of G by setting V_j to be the the union of U_w's with $u_w \in V_j'$. In each $G[V_j]$ we find a dominating set D_i with $\omega(D_j) = \gamma(G[V_j])$ and set $D := \bigcup_{j=1}^{k} D_j$.

Lemma 3.3 *Let (G, ω) be a vertex-weighted graph with $G \in \mathcal{C}_\rho$ and let D be the set obtained by* ApproxMWDS Phase 1. *Then*

$$\omega(D) \leq \gamma(G)/2.$$

Proof. Let $D^{(i)}$ denote the set in G after the ith iteration with $D^{(0)} := \bar{D}$. Let $P = (V_1, \ldots, V_k)$ be the vertex-weight partition of G obtained in the ith iteration. Since $D^{(i)} = \bigcup_{j=1}^{k} D_j^{(i)}$ where $D_j^{(i)}$ is a dominating set in $G[V_j]$, $D^{(i)}$ is a dominating set in G. Also, if $\bar{D}$ is a dominating set in G then every vertex from $V_j \setminus \partial(V_j)$ must be dominated by $D \cap V_j$. Consequently,

$$\omega(D_j^{(i)}) \leq \omega(\bar{D} \cap V_j) + \omega(\partial(V_j))$$

and so by Corollary 2.10 (a),

$$\omega(D^{(i)}) \leq \sum_{j=1}^{k} \omega(\bar{D} \cap V_j) + \omega(\partial(P)) \leq \omega(\bar{D}) + \omega(\partial(P)).$$

Since $\omega(\partial(P)) \leq \omega(V(G_{D^{(i-1)}}))/2$ and $\omega(V(G_{D^{(i-1)}})) = \omega(D^{(i-1)})$ we have

$$\omega(D^{(i)}) \leq \gamma(G) + \omega(D^{(i-1)})/2.$$

Thus, $\omega(D^{(i)}) \leq \gamma(G)\sum_{k=0}^{i} 2^{-k}$ and so after phase one $\omega(D) \leq 2\gamma(G)$. □
ApproxMWDS Phase 2. Let D be the dominating set obtained from ApproxMWDS Phase 1. Consider G_D and use the algorithm from Corollary 2.10 (b) with $p := d+1$ and $m = |G|$ to find a vertex-weight partition $(V_1', \ldots, V_k')$ of G_D. This gives a partition $P = (V_1, \ldots, V_k)$ of G as in phase one and we again find an optimal solution in each of $G[V_i]$'s and return the union.

Theorem 3.4 *Let $\mathcal{C}_\rho$ be a minor-closed family. There exists a distributed algorithm which given a positive integer d finds in vertex-weighted graph (G, ω) with $G \in \mathcal{C}_\rho$ a dominating set D with*

$$\omega(D) \leq \left(1 + \frac{1}{\log^d |G|}\right)\gamma(G).$$

The algorithm runs in a poly-logarithmic number of rounds.

Proof. Let D be the set obtained by ApproxMWDS Phase 1 and let D^* denote the set obtained from ApproxMWDS Phase 2. Then D^* is a dominating set and

$$\omega(D^*) \leq \gamma(G) + \omega(D)/\log^{d+1}|G| \leq \left(1 + \frac{1}{\log^d |G|}\right)\gamma(G).$$

The running time of the algorithm is poly-logarithmic as each $G[V_i]$ is poly-logarithmic in $m = |G|$ and so finding optimal solutions can be done in a poly-log number of rounds. □

3.3 Connected dominating sets

In [CH06c], in addition to distributed approximations for un-weighted versions of the previous two problems, we gave a distributed approximation for the unweighted version of the minimum connected dominating set problem. Unlike the maximum matching and the minimum dominating set problems, the weighted version of the minimum connected dominating set problem does not admit an efficient distributed algorithm. Consider a cycle $C = v_1, v_2, \ldots, v_L, v_1$ on $L \geq 2N$ vertices and suppose the identifiers of v_i's are in $\{1, \ldots, L\}$. We define three vertex-weighted graphs $(C, \omega_1), (C, \omega_2), (C, \omega_3)$ as follows. Let $M > 0$, we set $\omega_1(v_1) := M$ and $\omega_1(v_i) := 0$ if $i \neq 1$, $\omega_2(v_N) := M$ and $\omega_2(v_i) := 0$ if $i \neq N$, $\omega_3(v_1) := M, \omega_3(v_N) := M$ and $\omega_3(v_i) := 0$ if $i \neq 0, N$. Consider a distributed algorithm $\mathcal{A}$ for the minimum weight connected dominating set which runs in $T < N$ rounds. Let D_i denote the dominating set returned by $\mathcal{A}$ when run in (C, ω_i). In each execution of $\mathcal{A}$ each vertex v_i will use only the information about vertices within distance T to decide if it should or should not be included in the dominating set. Consequently, v_1 will have exactly the same information when $\mathcal{A}$ is run in (C, ω_1) and (C, ω_3). Similarly for v_N in (C, ω_2) and (C, ω_3). Clearly one of v_1, v_N must be in the dominating set D_3 for D_3 to induce a connected subgraph of C. If $v_1 \in D_3$ then $v_1 \in D_1$ and $\omega(D_1) = M$ while the weight of an optimal solution is 0. If $v_N \in D_3$ then $v_N \in D_2$ and $\omega(D_2) = M$.

Therefore, if a minor-closed family contains a sequence of connected graphs $G_1, G_2, \ldots, G_n, \ldots$ with finite circumferences $c(G_n)$ then no distributed approximation algorithm for the connected dominating set with a finite multiplicative error can run in $O(c(G_n))$ rounds. For example if $\mathcal{C}$ contains cycles of arbitrary large length then any distributed approximation algorithm for the minimum connected dominating set problem with a finite multiplicative error requires $\Omega(|G|)$ rounds for some G in $\mathcal{C}$.

References

[AGLP89] B. Awerbuch, A. V. Goldberg, M. Luby, S. A. Plotkin, Network Decomposition and Locality in Distributed Computation, Proc. 30th IEEE Symp. on Foundations of Computer Science (FOCS), (1989), 364–369.

[CH06a] A. Czygrinow, M. Hańćkowiak, Distributed algorithms for weighted problems in sparse graphs, *Journal of Discrete Algorithms*, Volume 4, (4), (2006), 588–607.

[CH06b] A. Czygrinow, M. Hańćkowiak, Distributed approximation algorithms in unit disc graphs, 20th International Symposium on Distributed Computing (DISC), LNCS 4167, (2006), 385–398.

[CH06c] A. Czygrinow, M. Hańćkowiak, Distributed almost exact approximations for minor-closed families, 14th Annual European Symposium on Algorithms (ESA), (2006), 244–255.

[CHS06] A. Czygrinow, M. Hańćkowiak, E. Szymańska, Distributed approximation algorithms in planar graphs, 6th Conference on Algorithms and Complexity (CIAC), LNCS 3998, (2006), 296–307.

[D97] R. Diestel,*Graph Theory*, 3rd Edition, Springer-Verlag, Berlin Heidelberg, (2005).

[E04] M. Elkin, An Overview of Distributed Approximation, in ACM SIGACT News Distributed Computing Column Volume 35, Number 4, 132, (2004), 40–57.

[KW03] F. Kuhn, R. Wattenhofer, Constant-Time Distributed Dominating Set Approximation, 22nd ACM Symposium on the Principles of Distributed Computing (PODC), (2003), 25 – 32.

[KMW04] F. Kuhn, T. Moscibroda, and R. Wattenhofer, What Cannot Be Computed Locally!, Proceedings of 23rd ACM Symposium on the Principles of Distributed Computing (PODC), (2004), 300–309.

[KMNW05] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, Fast Deterministic Distributed Maximal Independent Set Computation on Growth-Bounded Graphs, 19th International Symposium on Distributed Computing (DISC), (2005), 273–287.

[KMNW05] F. Kuhn, T. Moscibroda, T. Nieberg, and R. Wattenhofer, Local Approximation Schemes for Ad Hoc and Sensor Networks, 3rd ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC), (2005), 97–103.

[KP95] S. Kutten, D. Peleg, Fast distributed construction of k-dominating sets and applications, Proceedings of the 14th ACM symposium on Principles of Distributed Computing (PODC), (1995), 238–251.

[L92] N. Linial, Locality in distributed graph algorithms, *SIAM Journal on Computing*, 21 (1), (1992), 193–201.

[L86] M. Luby, A simple parallel algorithm for the maximal independent set problem, *SIAM Journal on Computing*, 15 (4), (1986), 1036–1053.

[NM05] J. Nesetril and P. Ossona de Mendez, Colorings and homomorphisms of minor closed classes, In B. Aronov, S. Basu, J. Pach, and M. Sharir, eds., Discrete and Computational Geometry, The Goodman-Pollack Festschrift, vol. 25 of Algorithms and Combinatorics, Springer, (2003), 651–664.