Discrete Applied Mathematics 170 (2014) 33–45

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Brushing without capacity restrictions

Darryn Bryant [a], Nevena Francetić [b], Przemysław Gordinowicz [c], David A. Pike [d], Paweł Prałat [e,*]

[a] The University of Queensland, Department of Mathematics, QLD 4072, Australia
[b] School of Mathematics & Statistics, Carleton University, Ottawa, ON, Canada
[c] Institute of Mathematics, Technical University of Lodz, Łódź, Poland
[d] Department of Mathematics and Statistics, Memorial University of Newfoundland, St. John's, NL, Canada
[e] Department of Mathematics, Ryerson University, Toronto, ON, Canada

ARTICLE INFO

Article history:
Received 19 November 2012
Received in revised form 22 January 2014
Accepted 30 January 2014
Available online 20 February 2014

Keywords:
Network cleaning
Brush number
Directed path covering

ABSTRACT

In graph cleaning problems, brushes clean a graph by traversing it subject to certain rules. Various problems arise, such as determining the minimum number of brushes that are required to clean the entire graph. This number is called the brushing number. Here, we study a new variant of the brushing problem in which one vertex is cleaned at a time, but more than one brush may traverse a dirty edge. In particular, we obtain results on the brushing number of Cartesian products of graphs and trees, as well as upper and lower bounds on the brushing number in the general case.

1. Introduction

In this paper, we study a natural variant of the graph cleaning problem that was introduced by McKeil [9] in which all edges and vertices of a graph are initially considered to be contaminated. Cleaning agents called brushes travel throughout the graph, decontaminating as they go. Once each vertex has been visited, and each edge has been traversed by a brush, the graph has then been cleaned, although in certain models recontamination may also occur. The model considered in Chapter 3 of [9] permits edges to be traversed by more than one brush at a time, and also permits edges to be traversed on multiple occasions.

In [10], restrictions are imposed whereby only dirty edges can be traversed, and each edge can be traversed by at most one brush. This model corresponds to the minimum total imbalance of the graph which is used in the graph drawing theory [4], and is well studied, especially when it is performed on random graphs [1,13,17]. (See also [8] for the algorithmic side, [11,16] for a related model of cleaning with Brooms, [12] for the relationship with other elimination schemes, and a combinatorial game [6].) In the present paper we relax one of these restrictions and allow dirty edges to be traversed by multiple brushes, although we retain the condition that an edge can be traversed on only one occasion.

Having been inspired by chip firing processes [2], the manner in which brushes disperse from an individual vertex is such that they do so in unison, provided that their vertex meets the criteria to fire. Models in which multiple vertices may fire simultaneously are called parallel cleaning models (see [5] for more details). In contrast, sequential parallel models mandate that vertices fire one at a time. The variant considered in [10] and the one we consider in this paper are sequential in nature.

* Corresponding author. Tel.: +1 416 979 5000; fax: +1 416 598 5917.
E-mail addresses: db@maths.uq.edu.au (D. Bryant), nfrancet@uottawa.ca (N. Francetić), pgordin@p.lodz.pl (P. Gordinowicz), dapike@mun.ca (D.A. Pike), pralat@ryerson.ca (P. Prałat).

http://dx.doi.org/10.1016/j.dam.2014.01.024
0166-218X/

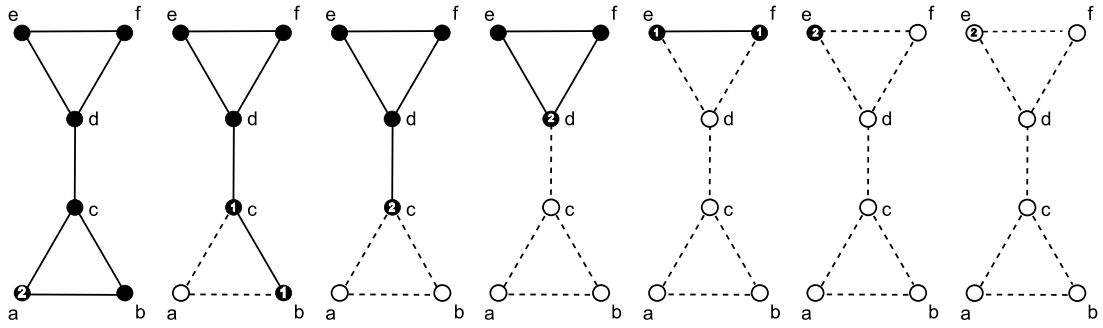

Fig. 1. A graph G with an initial configuration of 2 brushes.

When considering a graph or network, the central question under investigation is that of determining the brush number for the graph (i.e., the minimum number of brushes that enable the graph to be cleaned), as well as to describe a corresponding cleaning strategy. Subsequent to presenting a more formal definition of our model in Section 2, we establish some general bounds for the brush number of an arbitrary graph, including bounds that are expressed in terms of parameters such as cutwidth and bisection width. We then investigate two specific classes of graphs, namely Cartesian products and trees.

For Cartesian products we prove a general upper bound on the brush number, and then establish exact values for the brush numbers of m by n grids and hypercubes. For trees, we prove that if a tree T has $d_\ell(T)$ vertices of degree 1, then exactly $(d_\ell(T) + 1)/2$ brushes are required for an optimal cleaning strategy when $d_\ell(T)$ is odd. If $d_\ell(T)$ is even, then the minimum number of brushes that are required is either $d_\ell(T)/2$ or $d_\ell(T)/2 + 1$, which is to say that trees with an even number of vertices of degree 1 are partitioned into two sets depending on whether their brush number equals $d_\ell(T)/2$ or exceeds it by 1.

2. Definitions

As already noted, the graph cleaning model we consider here differs from the one presented in [10] in that we allow edges to be traversed by multiple brushes. Before we define the model rigorously, we present a simple example illustrated in Fig. 1.

In this example, all edges and vertices are initially dirty and we place two brushes at vertex a. As vertex a contains at least as many brushes as dirty incident edges, it is able to fire. Thus, at Step 1 vertex a is cleaned and a brush is sent down each dirty edge: edges ab and ac have been cleaned; vertices b and c have one fewer dirty incident edges. At Step 2, vertices c, d, e and f cannot be cleaned as they each have fewer brushes than dirty incident edges; vertex b is cleaned instead and one brush is sent to c. At Step 3, the only vertex ready to be cleaned is vertex c. It is at this point that the cleaning process considered in this paper differs from the process described in [10]. In [10], only one brush is permitted to traverse an edge. So one brush would be moved from c to d, whilst the other would remain at c. With our variant of the cleaning process, more than one brush can be moved through a dirty edge. Thus, the two brushes are moved from c to d (no advantage can be gained by leaving a brush behind). At Step 4, d is cleaned and the remaining two vertices (e and f) can be cleaned at the next two steps (although all edges are clean prior to the final step). Under our cleaning model this example graph can be cleaned with just two brushes, whereas under the model considered in [10] three brushes would be required.

Now we formally define the cleaning process we are considering. Let $G = (V, E)$ be any finite, undirected graph. The initial configuration of brushes is given by the function $\omega_0 : V \to \mathbb{N}_0$, where $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$, $\omega_0(v)$ is the number of brushes initially at vertex v, and all vertices and edges of the graph are initially dirty. At each step t of the process, $\omega_t(v)$ denotes the number of brushes at vertex $v \in V$, and $D_t \subseteq V$ denotes the set of dirty vertices. An edge $uv \in E$ is dirty if and only if both u and v are dirty; that is, $\{u, v\} \subseteq D_t$. Finally, let $D_t(v)$ denote the number of dirty edges incident to v at step t; that is,

$$D_t(v) = \begin{cases} |N(v) \cap D_t| & \text{if } v \in D_t \\ 0 & \text{otherwise} \end{cases}$$

(where $N(v)$ denotes, as usual, the neighbourhood of v).

Definition 2.1. The **(generalized) cleaning process** $\mathfrak{P}(G, \omega_0) = \{(\omega_t, D_t)\}_{t=0}^{T}$ of an undirected graph $G = (V, E)$ with an **initial configuration of brushes** ω_0 is as follows:

⓪ Initially, all vertices are dirty: $D_0 = V$; set $t = 0$.
① Let α_{t+1} be any vertex in D_t such that $\omega_t(\alpha_{t+1}) \geq D_t(\alpha_{t+1})$. If no such vertex exists, then stop the process (set $T = t$), return the **cleaning sequence** $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$, the **final set of dirty vertices** D_T, the **final configuration of brushes** ω_T, and the **distribution of brushes** δ.

② Assign to each dirty edge $v\alpha_{t+1}$ incident to α_{t+1} a natural number $\delta(v\alpha_{t+1})$ such that the sum of numbers assigned is at most $\omega_t(\alpha_{t+1})$; in other words, for each dirty neighbour v of α_{t+1}, let $\delta(v\alpha_{t+1}) \in \mathbb{N}$ and

$$\rho(\alpha_{t+1}) = \sum_{v \in N(\alpha_{t+1}) \cap D_t} \delta(v\alpha_{t+1}) \leq \omega_t(\alpha_{t+1}). \tag{1}$$

③ Clean α_{t+1} and all dirty incident edges by moving $\delta(v\alpha_{t+1})$ brushes from α_{t+1} to each dirty neighbour v of α_{t+1}. Consequently, $D_{t+1} = D_t \setminus \{\alpha_{t+1}\}$, $\omega_{t+1}(\alpha_{t+1}) = \omega_t(\alpha_{t+1}) - \rho(\alpha_{t+1})$, and for every $v \in N(\alpha_{t+1}) \cap D_t$, $\omega_{t+1}(v) = \omega_t(v) + \delta(v\alpha_{t+1})$, otherwise $\omega_{t+1}(v) = \omega_t(v)$.
④ Set $t = t + 1$ and go back to ①.

The function δ assigns the number of brushes that traverse a dirty edge. In the model considered in [10], exactly one brush could be moved through each dirty edge; that is, $\delta(uv) = 1$ for each $uv \in E$ (provided that the graph has been cleaned, otherwise $\delta(uv) = 1$ for each pair of clean vertices). We also mentioned that there is no advantage of leaving a brush behind; hence we may always assume that $\rho(\alpha_{t+1}) = \omega_t(\alpha_{t+1})$, provided that there is at least one dirty edge incident to α_{t+1} at the time it is cleaned (that is, provided that $N(\alpha_{t+1}) \cap D_t \neq \emptyset$). In other words, in such situations, the inequality in (1) is, in fact, equality.

It was shown in [10] that, given a graph G and an initial configuration ω_0, the cleaning process of [10] returns a unique final set of dirty vertices. Consequently, if there exists a cleaning sequence that cleans G, we know that every (legal) cleaning sequence will clean G. Such is not the case here; whether a graph is cleaned entirely or not may depend on the function δ (rather than just on the initial configuration ω_0 of brushes). However, it still makes sense to ask for the minimum number of brushes distributed on the vertices of a given graph G having the initial configuration of brushes ω_0 (this time, together with the function δ and cleaning strategy) needed to clean G.

Definition 2.2. A graph $G = (V, E)$ **can be cleaned** by the initial configuration of brushes ω_0 if there is a cleaning process $\mathfrak{P}(G, \omega_0)$ that returns an empty final set of dirty vertices (that is, $D_T = \emptyset$).

Let the **(generalized) brush number** of G, denoted $B(G)$, be the minimum number of brushes needed to clean G; that is,

$$B(G) = \min_{\omega_0 : V \to \mathbb{N}_0} \left\{ \sum_{v \in V} \omega_0(v) : G \text{ can be cleaned by } \omega_0 \right\}.$$

The brush number for the model in [10] was denoted by $b(G)$. Since there were more restrictions for this process than for the generalized one which we are considering, we immediately get that $B(G) \leq b(G)$. As we will see, the difference can be arbitrarily large although there are some classes of graphs for which $B(G) = b(G)$.

When a graph G is cleaned using the cleaning process, each edge of G is traversed by at least one brush. Note that no brush may return to a vertex it has already visited, motivating the following definition.

Definition 2.3. The **brush path** of a given brush is the path formed by the set of edges cleaned by that brush.

By definition, G can be covered by $B(G)$ brush paths. The minimum number of (undirected) paths by which a graph G can be covered therefore yields a lower bound for $B(G)$. This is only a lower bound because some path covers would not be valid in the cleaning process. We will present such examples soon. However, these paths can be treated as directed paths (with their orientation corresponding to the chronological direction of the brushes), which gives us another way of thinking about this problem.

Lemma 2.4. *$B(G)$ is the minimum number of directed paths in G whose union is acyclic and covers the edges of G (where acyclic means no directed cycles and where the two arcs (u, v) and (v, u) are considered to be a cycle).*

Proof. One direction is obvious. If the graph can be cleaned with b brushes, then each brush yields a directed path. At the end of the process every edge is clean, so all edges are covered by b paths. It remains to show that the union is acyclic. For a contradiction, suppose that there is a directed cycle $v_1, v_2, \ldots, v_k$. It follows that v_2 must have been cleaned after v_1, v_3 after v_2, and so on. Finally, we deduce that v_1 must have been cleaned after v_k and we get a contradiction we were hoping for.

In order to prove the other direction, suppose that there exist b directed paths in G whose union is acyclic and covers the edges of G. We want to show that the graph can be cleaned with b brushes and at the end of the process the paths we started with are yielded. We start with putting one brush at the beginning of each directed path (there are b brushes in total). Which vertex should be cleaned first? Let us take any vertex $v \in V(G)$. If v has no in-neighbour, it can be cleaned first. Otherwise, we pick any in-neighbour of v and continue investigation. Since the graph is acyclic, the process must finish successfully which guarantees that there is a vertex in G that has no in-neighbour. After cleaning this vertex, we continue the selection process the same way (applied to the graph with the vertices already cleaned removed) until the whole graph is cleaned. □

Finally, let us mention that the process of [10] forces each edge to be traversed by exactly one brush and so G can be decomposed into $b(G)$ brush paths. This yields a lower bound for $b(G)$, although this bound is often not tight. For example, K_4 can be decomposed into 2 undirected paths but $b(K_4) = 4$.

3. Some observations and upper bounds

In this section, in order to warm up, we make some simple but useful observations. Suppose first that a graph G on n vertices (together with an initial configuration of brushes) can be cleaned using the cleaning sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$. In [10] it was shown that the cleaning process can be reversed; that is, the final configuration of brushes can be used to re-clean the graph using the cleaning sequence $\alpha^{-1} = (\alpha_n, \alpha_{n-1}, \ldots, \alpha_1)$. This is also the case for the cleaning variant we study in this paper.

Theorem 3.1 (*The Reversibility Theorem*). *Given an initial configuration ω_0, suppose G can be cleaned yielding final configuration ω_n, $n = |V(G)|$. Then, given initial configuration $\tau_0 = \omega_n$, G can be cleaned yielding the final configuration $\tau_n = \omega_0$.*

Proof. Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ be a cleaning sequence (together with the assignment and cleaning strategy) of the cleaning process $\mathfrak{P}_+ = \{(\omega_t, D_t)\}_{t=0}^n$ which cleans the graph G. Let $E^-(\alpha_t) = \{\alpha_t\alpha_i \in E(G) : i < t\}$ be the set of clean edges incident to α_t when this vertex is about to be cleaned, and similarly let $E^+(\alpha_t) = \{\alpha_t\alpha_i \in E(G) : i > t\}$ be the set of dirty edges. Clearly $\deg(\alpha_t) = |E^-(\alpha_t)| + |E^+(\alpha_t)|$. Finally, let $\delta(e)$, $e \in E(G)$ be the assignment of brushes that is done during the cleaning process $\mathfrak{P}_+$.

It is straightforward to see that for every t ($1 \le t \le n$)

$$\omega_n(\alpha_t) = \omega_0(\alpha_t) + \sum_{e\in E^-(\alpha_t)} \delta(e) - \sum_{e\in E^+(\alpha_t)} \delta(e). \tag{2}$$

($\sum_{e\in E^-(\alpha_t)} \delta(e)$ is the number of brushes sent to α_t; $\sum_{e\in E^+(\alpha_t)} \delta(e)$ is the number of brushes sent from α_t.) Now, we will show that the cleaning process $\mathfrak{P}_- = \{(\tau_t, C_t)\}_{t=0}^n$, $\tau_0 = \omega_n$, can be used to clean G using a cleaning sequence $\alpha^{-1} = (\alpha_n, \alpha_{n-1}, \ldots, \alpha_1)$; that is, vertex α_{n-t+1} is cleaned at time t. We will show that exactly the same assignment of brushes, $\delta(e)$, can be achieved for the reversed process $\mathfrak{P}_-$ as for $\mathfrak{P}_+$. For that, we use induction on t. Since $E^+(\alpha_n) = \emptyset$, it follows from (2) that

$$\tau_0(\alpha_n) = \omega_n(\alpha_n) = \omega_0(\alpha_n) + \sum_{e\in E^-(\alpha_t)} \delta(e) \ge \sum_{e\in E^-(\alpha_t)} \delta(e).$$

This implies that the vertex α_n can be cleaned at the first step and the required assignment can be achieved. The basis step is verified.

For the induction step, assume that vertices $\alpha_n, \alpha_{n-1}, \ldots, \alpha_{k+1}$ ($1 \le k < n$) are clean at the beginning of time-step $n - k + 1$ (or, equivalently, at the end of time-step $n - k$) of the process $\mathfrak{P}_-$ and that α_k received $\sum_{e\in E^+(\alpha_k)} \delta(e)$ extra brushes from its neighbours. It is not difficult to check, again using (2), that α_k can be cleaned in this step. Indeed,

$$\tau_{n-k}(\alpha_k) = \tau_0(\alpha_k) + \sum_{e\in E^+(\alpha_k)} \delta(e) = \omega_0(\alpha_k) + \sum_{e\in E^-(\alpha_k)} \delta(e) \ge \sum_{e\in E^-(\alpha_k)} \delta(e),$$

and so the vertex α_k can be cleaned and the required assignment can be achieved as well. This finishes the proof of the theorem. □

As $B(G) \le b(G)$ any upper bound for $b(G)$ serves as an upper bound for $B(G)$. We will show in Section 5.2 that $b(G) - B(G)$ can be arbitrarily large but there are families of graphs for which the values are the same. Indeed for any $n \in \mathbb{N}$ we have $B(C_n) = b(C_n) = 2$ where C_n is a cycle of length n, $B(P_n) = b(P_n) = 1$ where P_n is a path of length $n - 1$, and $B(K_n) = b(K_n) = \lfloor n^2/4 \rfloor$ where K_n is a clique of order n (note that one needs to introduce $\max\{n - 2t - 1, 0\}$ brushes in order to clean a vertex at time $t \in \mathbb{N}$). For such families, upper bounds for $b(G)$ turn out to be very useful.

Let us mention one bound that has been used a number of times and which might also be useful for $B(G)$. From Theorem 3.7 in [1] it follows that

$$B(G) \le b(G) \le \frac{|E|}{2} + \frac{|V|}{4} - \frac{1}{4} \sum_{v\in V(G), \deg(v) \text{ is even}} \frac{1}{\deg(v) + 1}$$

for any graph $G = (V, E)$. The proof of this result is non-constructive and the bound is sharp, for example, for cliques.

4. Relation to the cutwidth and the bisection width

In this section, we investigate relations between the brush number, the cutwidth, and the bisection width of graphs. In the next sections, we will use these relations to establish bounds on the brush number.

4.1. Lower bound

We start with a definition of the cutwidth.

Definition 4.1. Let $G = (V, E)$ be a graph on n vertices and let $f : V \to \{1, 2, \ldots, n\}$ be a bijection; that is, f is a **linear layout** of G. The **cutwidth** of f (with respect to G) is defined as follows:

$$cw_f(G) = \max_{1 \le i \le n} |\{uv \in E : f(u) \le i < f(v)\}|.$$

The **cutwidth** of G, $cw(G)$, is the minimum cutwidth over all possible linear layouts of G; that is, $cw(G) = \min_f cw_f(G)$.

In [10] it was observed that the cutwidth serves as a lower bound for $b(G)$. We show in Theorem 4.2 that cutwidth is also a lower bound for $B(G)$. This lower bound is sometimes sharp (see, for example, the result for hypercubes (Theorem 5.5)), but is sometimes very weak (consider, for example, a collection of n disjoint copies of K_3: $B(nK_3) = 2n$ whereas $cw(nK_3) = 2$).

Theorem 4.2. *For any graph G, $B(G) \ge cw(G)$.*

Proof. The proof is straightforward. Let $G = (V, E)$ be any graph on n vertices, and consider the cleaning sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ that yields $B(G)$. Note that α can be viewed as the linear layout $f_\alpha : V \to \{1, 2, \ldots, n\}$ defined as follows: $f_\alpha(\alpha_i) = i$ for $i \in \{1, 2, \ldots, n\}$. To get the lower bound for $B(G)$, for any $i \in \{1, 2, \ldots, n\}$, let

$$\gamma_i = \{uv \in E : f_\alpha(u) \le i < f_\alpha(v)\} = \{\alpha_x\alpha_y \in E : x \le i < y\},$$

and note that at the end of time-step i of the process, at least one brush traversed every edge from γ_i. Moreover, every edge from γ_i was traversed by different brushes, as one end of the edge is still dirty at this point. Therefore, at least $|\gamma_i|$ brushes are needed to clean the graph G. The result follows, since the claim is true for any i and so

$$B(G) \ge \max_{1 \le i \le n} |\gamma_i| = cw_{f_\alpha}(G) \ge cw(G).$$

The proof is finished. □

Theorem 4.2 has a useful corollary, but before we state it we need one more definition.

Definition 4.3. Let $G = (V, E)$ be any graph with $|V| = n$. For $S \subseteq V$, let $e(S, V \setminus S)$ denote the number of edges between S and its complement. A **bisection** of V is a partition of V into two parts, S and $V \setminus S$, such that $|S| = \lfloor n/2 \rfloor$ and $|V \setminus S| = \lceil n/2 \rceil$. The **size of a bisection** $(S, V \setminus S)$ is the number of edges crossing between the parts; that is, $e(S, V \setminus S)$. A **minimum bisection** is a bisection of V with minimal size. Finally, the size of a minimum bisection is called the **bisection width** and denoted by $bw(G)$.

Corollary 4.4. *Let $G = (V, E)$ be any graph on n vertices. For any $k \in \{1, 2, \ldots, n\}$, let*

$$B_k = \min_{S \subseteq V, |S| = k} e(S, V \setminus S).$$

Then, $B(G) \ge \max_k B_k$. In particular, $B(G) \ge b_{\lfloor n/2 \rfloor} = bw(G)$.

Proof. It is enough to notice that for every layout f of G we have the following

$$cw_f(G) = \max_{1 \le k \le n} |\{uv \in E : f(u) \le k < f(v)\}| \ge \max_{1 \le k \le n} B_k,$$

and so the assertion holds. □

4.2. Upper bound

The cutwidth of G, $cw(G)$, that served as a lower bound in the previous subsection is the minimum cutwidth over all possible linear layouts of G. It turns out that a more restricted version is a convenient, and sometimes sharp, upper bound for the brush number.

Definition 4.5. The **restricted cutwidth** of a graph $G = (V, E)$ on n vertices, $cw_p(G)$, is the minimum cutwidth over all possible linear layouts of G that induce a path; that is,

$$cw_p(G) = \min\left\{cw_f(G) : f^{-1}(i)f^{-1}(i+1) \in E \text{ for all } i \in \{1, 2, \ldots, n-1\}\right\},$$

provided that the graph G has a Hamiltonian path. Otherwise, $cw_p(G) = \infty$.

Before we move back to the brush number, note that for every graph G, $cw(G) \le cw_p(G)$ by the definition of restricted cutwidth.

Theorem 4.6. *For any graph G, $B(G) \le cw_p(G)$.*

Proof. If G is not a Hamiltonian graph, the claim is trivial. Otherwise, let f be a linear layout of G that yields $cw_p(G)$. We clean G using the cleaning sequence associated with the layout f. The edges $f^{-1}(i)f^{-1}(i+1)$, $i = 1, 2, \ldots, n-1$, of the Hamilton path induced by f are called *special* edges, and the other edges of G are called *normal* edges. The cleaning strategy is as follows. Start with $\hat{b} := cw_p(G) = cw_f(G)$ brushes at the vertex $f^{-1}(1)$. At every step of the process, send exactly one brush through each normal, dirty edge; the remaining brushes are sent through the special, dirty edges that form a backbone of the graph G. It is clear that this strategy cleans the graph. At the end of a given time-step t, the number of edges between clean vertices and dirty ones is $|\gamma_t|$, where

$$\gamma_t = \{uv \in E : f(u) \leq t < f(v)\}.$$

Hence, the number of normal edges of this type is $|\gamma_t| - 1$, which is also the number of brushes that traversed them. All other brushes were available at the beginning of time-step t and so the number of brushes that were sent through the special edge $f^{-1}(t)f^{-1}(t+1)$ is equal to $\hat{b} - (|\gamma_t| - 1) \geq 1$. The process can be continued and the proof is finished. □

5. Cartesian products

In this section, we provide a universal upper bound for the Cartesian product of two graphs. After that we apply this result to two families of graphs: grids and hypercubes. We start with a definition of the Cartesian product.

Definition 5.1. The graph $G\square H$ is the **Cartesian product** of graphs G and H. It consists of the vertex set $V(G) \times V(H)$ where $(u, v) \in V(G\square H)$ is adjacent to $(u', v') \in V(G\square H)$ when either $u = u'$ and $vv' \in E(H)$ or $v = v'$ and $uu' \in E(G)$.

5.1. General upper bound

It can be easily observed that $G\square H$ decomposes into $|V(G)|$ copies of H or $|V(H)|$ copies of G. This property is used in creating an upper bound for $B(G\square H)$. It was shown in [10] that

$$b(G\square H) \leq |V(H)|b(G) + |V(G)|b(H),$$

and so this bound can be used as an upper bound for $B(G)$ as well. However, an upper bound which is often much stronger can be obtained for $B(G)$.

Theorem 5.2. *For any two graphs G and H,*

$$B(G\square H) \leq cw_p(G\square H) \leq |V(G)|cw_p(H) + cw_p(G). \tag{3}$$

Before we give the proof of this theorem, note that by symmetry we also have

$$B(G\square H) \leq cw_p(G\square H) \leq |V(H)|cw_p(G) + cw_p(H), \tag{4}$$

so one can use (3) or (4), depending which bound gives a stronger result.

Proof. We will show that $cw_p(G\square H) \leq |V(G)|cw_p(H) + cw_p(G)$, from which the result follows since $B(G\square H) \leq cw_p(G\square H)$ (see Theorem 4.6). Note that if either G or H has no Hamiltonian path, then by definition we have $|V(G)|cw_p(H) + cw_p(G) = \infty$ and the result holds. Thus, we assume that G and H both have Hamiltonian paths.

Let $(v_1, v_2, \ldots, v_n)$, $n = |V(G)|$, be a Hamiltonian path in G induced by the linear layout g which yields $cw_p(G)$. Also, let $(w_1, w_2, \ldots, w_k)$, $k = |V(H)|$, be a Hamiltonian path in H induced the linear layout h of H that yields $cw_p(H)$. Consider the Cartesian product of G and H. Let us colour edges of each copy of H *red* and edges of each copy of G *green*. Copies of G are ordered according to the layout h (that is, the ith copy of G, G_i, has vertices $(v_1, w_i), (v_2, w_i), \ldots, (v_n, w_i)$). In order to provide an upper bound of $cw_p(G\square H)$ we consider the following layout of $G\square H$. Use g applied to G_1, then the inverse of g (that is Hamiltonian path $(v_n, v_{n-1}, \ldots, v_1)$) applied to G_2, then g again but applied to G_3, etc. (See Fig. 2 in which dashed edges outline Hamiltonian paths.)

Formally, take the following Hamiltonian path in $G\square H$:

$$\begin{aligned}&((v_1, w_1), (v_2, w_1), \ldots, (v_n, w_1),\\ &(v_n, w_2), (v_{n-1}, w_2), \ldots, (v_1, w_2),\\ &(v_1, w_3), (v_2, w_3), \ldots, (v_n, w_3), \ldots),\end{aligned}$$

and denote by f the linear layout of $G\square H$ which induces this path. Then, for every s $(1 \leq s \leq nk)$ we have

$$\left|\{(v_{j_1}, w_{i_1})(v_{j_2}, w_{i_2}) \in E : f((v_{j_1}, w_{i_1})) \leq s < f((v_{j_2}, w_{i_2}))\}\right| \leq n \cdot cw_p(H) + cw_p(G),$$

since there are always at most $cw_p(G)$ green edges and at most $n \cdot cw_p(H)$ red edges. For example, Fig. 3 gives a linear layout of the graph $G\square H$ from Fig. 2. Here, the solid edges are the edges of the copies of G (called 'green' in the proof) and the dashed edges are the edges of the copies of H (called 'red'). The vertical line corresponds to the restricted cutwidth of this layout. In particular, there are $|V(G)|cw_p(H)$ edges of H and $cw_p(G)$ edges of G crossing the vertical line.

Hence $cw_p(G\square H) \leq cw_f(G\square H) \leq n \cdot cw_p(H) + cw_p(G)$. (In fact, $cw_f(G\square H) = n \cdot cw_p(H) + cw_p(G)$.) The proof of the theorem is finished. □

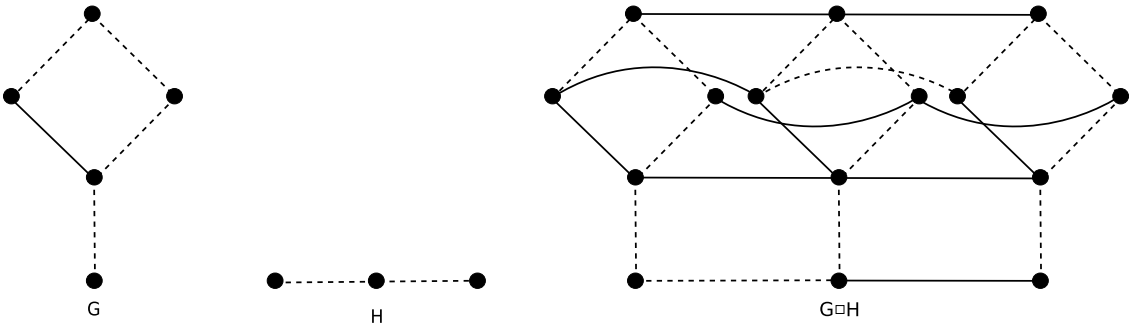

Fig. 2. The layout used to clean $G\square H$.

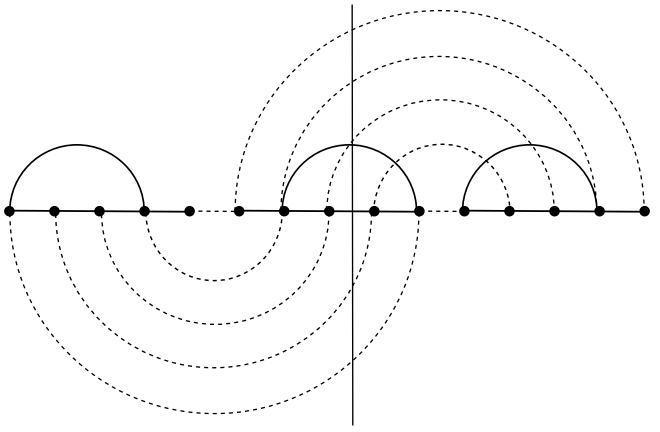

Fig. 3. The layout used to clean $G\square H$.

5.2. Grids

In this subsection, we investigate grids; that is, Cartesian products of two finite paths. The bisection width of $P_n\square P_n$ is well studied [3,7,15]. It is known that for every $n \geq 2$,

$$B_{\lfloor n^2/2 \rfloor} = bw(P_n\square P_n) = \begin{cases} n & \text{if } n \text{ is} \\ n+1 & \text{otherwise.} \end{cases}$$

(B_k was introduced in Corollary 4.4.) For odd n, this will be enough to establish a lower bound on $B(P_n\square P_n)$. For even $n \geq 4$, we note that $B_{(n^2/2)-1} = B_{(n^2/2)+1} = n+1$. In both cases, it follows from Corollary 4.4 that $B(P_n\square P_n) \geq n+1$. To get a matching upper bound we simply use Theorem 5.2, since $cw_p(P_n) = 1$.

Theorem 5.3. *For every $n \in \mathbb{N} \setminus \{1, 2\}$, $B(P_n\square P_n) = n+1$.*

Note that this result can be extended to Cartesian products of non-isomorphic paths. For any $m > n \geq 2$, $B(P_m\square P_n) = n+1$. Let us also mention that grids can be used to show that $b(G) - B(G)$ could be any non-negative integer and that the ratio $b(G)/B(G)$ can be arbitrarily close to any $c \in [2, \infty)$. Indeed, the classic brush number was studied in [10] and for $n \in \mathbb{N} \setminus \{1, 2\}$ we have

$$b(P_n\square P_n) - B(P_n\square P_n) = (2n-2) - (n+1) = n-3.$$

On the other hand, if $m = \lceil an \rceil$ for some $a \in [1, \infty)$, then

$$\frac{b(P_m\square P_n)}{B(P_m\square P_n)} = \frac{m+n-2}{n+1} \to a+1,$$

as $n \to \infty$.

Next, we give an example where $b(G)/B(B)$ is arbitrarily close to any $c \in [1, 2]$. Let $m = \lceil an \rceil \geq n$ for some $a \in [1, \infty)$ and let G be a graph obtained from two complete graphs, K_n and K_m, by connecting them by an edge. It is straightforward to see that

$$B(G) = B(K_m) = b(K_m) = \lfloor m^2/4 \rfloor = m^2/4 + O(1)$$

(we can first clean K_m and then reuse the brushes to clean K_n), and that

$$b(G) = b(K_m) + b(K_n) + O(1) = \lfloor m^2/4 \rfloor + \lfloor n^2/4 \rfloor + O(1) = (m^2 + n^2)/4 + O(1)$$

(since adding an edge to any graph can change the value of the brush number by at most one). Hence,

$$\frac{b(G)}{B(G)} = \frac{(m^2 + n^2)/4 + O(1)}{m^2/4 + O(1)} \to 1 + \frac{1}{a^2},$$

as $n \to \infty$.

5.3. Hypercubes

As usual, we start with a definition.

Definition 5.4. The vertices of a hypercube Q_n are points in an n-dimensional space over the field with two elements $F_2 = \{0, 1\}$. Two points are adjacent in Q_n if and only if they differ in exactly one coordinate.

We have $Q_1 = P_2$, $Q_2 = C_4$, and Q_3 is the cube in 3-dimensional space. For every $n \in \mathbb{N}$ we have $Q_{n+1} = Q_1 \square Q_n$.

In [14] it was shown that $cw(Q_n) = \frac{2}{3}(2^n - a)$ where $a = 1$ if n is even and $a = \frac{1}{2}$ otherwise. From Theorem 4.2 we have that $B(Q_n) \geq cw(G)$ giving the lower bound. We will show that the restricted cutwidth is exactly the same which implies the following result.

Theorem 5.5. *For every $n \in \mathbb{N}$, $B(Q_n) = r_n$, where*

$$r_n = \begin{cases} \frac{2}{3}(2^n - 1) & \text{if } n \text{ is even} \\ \frac{2}{3}\left(2^n - \frac{1}{2}\right) & \text{otherwise.} \end{cases}$$

Proof. As we already mentioned, it is enough to show that $cw_p(Q_n) \leq r_n$ for every $n \in \mathbb{N}$. The proof is by induction on n. Note that $cw_p(Q_1) = cw_p(K_2) = 1$ so the result holds for $n = 1$. For every $n \geq 2$, it follows from Theorem 5.2 that

$$cw_p(Q_n) = cw_p(Q_{n-1} \square K_2) \leq 2cw_p(Q_{n-1}) + 1,$$

since $cw_p(K_2) = 1$. We will show that it is possible to improve it slightly if n is even; that is, we will show that for even n,

$$cw_p(Q_n) \leq 2cw_p(Q_{n-1}).$$

This will finish the proof, since

$$r_n = \begin{cases} 2r_{n-1} + 1 & \text{if } n \text{ is odd} \\ 2r_{n-1} & \text{otherwise,} \end{cases}$$

and so $cw_p(Q_n) \leq r_n$ for every $n \in \mathbb{N}$.

Fix any even $n \in \mathbb{N}$, and consider a layout f that yields $cw_p(Q_n)$. It is enough to show that $cw_p(Q_n)$ is even. In fact, we will show that $|\gamma_t|$ is even for every t, where

$$\gamma_t = \{uv \in E : f(u) \leq t < f(v)\}.$$

Since every vertex has degree n, $|\gamma_1| = n$ is even. Suppose now that $|\gamma_t|$ is even and that there are k edges from vertices with $f(u) < t + 1$ to the vertex v with $f(v) = t + 1$. It follows that there are $n - k$ edges from the vertex v with $f(v) = t + 1$ to vertices with $f(u) > t + 1$. Hence,

$$|\gamma_{t+1}| = |\gamma_t| - k + (n - k) = |\gamma_t| - 2k + n,$$

which is even, since each term is even. The proof is complete. □

6. Trees

In this section we turn our attention towards trees, which appear to be, perhaps surprisingly, non-trivial to analyse. Since every leaf of a given tree T must start or finish at least one brush path, then clearly $B(T) \geq \lceil d_\ell(T)/2 \rceil$ where $d_\ell(T)$ denotes the number of leaves of T.

One insight that will eventually help us to establish an upper bound on $B(T)$ is that any cleaning strategy for a tree can be easily turned into a strategy in which each brush starts and ends at a leaf. Similar to Lemma 2.4, there is a natural bijection between such a cleaning strategy with t brushes and an oriented path **covering** of the tree that has t directed paths such

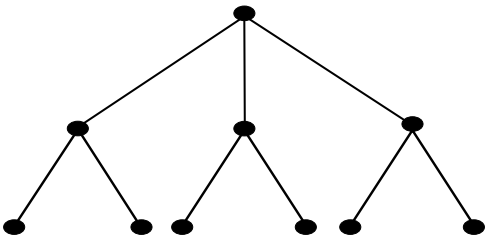

Fig. 4. The smallest example of a tree with $B(T) = d_\ell(T)/2 + 1$.

that the end-vertices of each directed path in the covering are vertices of degree 1 in the tree and all paths which cover the same edge are consistently oriented along that edge.

Since any oriented path covering immediately yields a covering with undirected paths (by simply disregarding the orientation of the directed covering), we could attempt to prove that there are trees with large brush number by showing that there are some trees which cannot be covered by $\lceil d_\ell(T)/2 \rceil$ undirected paths. However, it turns out that $\lceil d_\ell(T)/2 \rceil$ paths always suffice.

Theorem 6.1. *For every tree T with $d_\ell(T)$ leaves, the edges of T can be covered with $\lceil d_\ell(T)/2 \rceil$ undirected paths.*

Proof. Arbitrarily choose a set $\mathcal{P}$ of $\lceil d_l(T)/2 \rceil$ paths in T such that the ends of each path are leaves, and such that every leaf is the end of some path in $\mathcal{P}$. Suppose e is an edge that is not covered by $\mathcal{P}$, and let P be a path containing e such that the ends of P are leaves. Clearly, P exists and $P \notin \mathcal{P}$. Let $P', P'' \in \mathcal{P}$ such that P' contains one end of P and P'' contains the other. It is easy to see that the edges of $E(P) \cup E(P') \cup E(P'')$ can be covered by just two paths in T (if x' and y' are the ends of P', and x'' and y'' are the ends of P'', then the unique path from x' to x'' and the unique path from y' to y'' can be used). Replace P' and P'' with these two paths and repeat this process until all the edges of T are covered. □

Given that there are no trees for which an undirected path covering with $\lceil d_\ell(T)/2 \rceil$ paths does not exist, it is tempting to speculate that $\lceil d_\ell(T)/2 \rceil$ might be the actual value of $B(T)$ for every tree T. However, there do exist examples of trees with an even number of leaves and a brush number of $d_\ell(T)/2 + 1$ (see Fig. 4 for the smallest example, which has 6 leaves). As it happens, the worst scenario possible is that the brush number of T exceeds $d_\ell(T)/2$ by just 1. The rest of this section is devoted to proving the following result.

Theorem 6.2. *Let T be any tree with $d_\ell(T)$ leaves. Then, the following hold.*

(i) *If $d_\ell(T)$ is odd, then*

$$B(T) = \frac{d_\ell(T) + 1}{2}.$$

(ii) *If $d_\ell(T)$ is even, then*

$$\frac{d_\ell(T)}{2} \leq B(T) \leq \frac{d_\ell(T)}{2} + 1.$$

In order to prove the theorem we need several lemmata that we state next (their proofs will follow in due course). The first one implies that in order to prove the main theorem it is enough to do so for trees in which every internal vertex has degree 3. Let $\mathcal{T}$ denote this family of trees.

Lemma 6.3. *Let T be any tree with $d_\ell(T)$ leaves and let $B(T)$ be its brush number. There exists a tree $T^* \in \mathcal{T}$ such that $d_\ell(T) = d_\ell(T^*)$ and $B(T) \leq B(T^*)$.*

It follows from Lemma 6.3 that if there exists a counterexample T to one of the two assertions (i) and (ii) stated in Theorem 6.2, then there is a counterexample T^* from the family $\mathcal{T}$ with the same number of degree 1 vertices as T.

Before we present the second lemma, we need to make some observations. Consider a tree $T \in \mathcal{T}$ with at least two vertices (that is, $T \in \mathcal{T} \setminus \{K_1\}$). Select any vertex of degree 1 and call it the **special leaf**; the remaining $k = d_\ell(T) - 1$ vertices of degree 1 will be called **ordinary leaves**. We say that T is a rooted at the special leaf. We want to clean the tree T starting with m brushes occupying exactly m ordinary leaves and, perhaps, some brushes placed at the special leaf. If no brush gets stuck at some internal vertex and each leaf either starts or finishes one brush path, then we say that an **optimal** cleaning can be achieved. In such a situation, $k - m$ ordinary leaves finish with exactly one brush and so the number of brushes that will end at the special leaf is

$$r(k, m) = m - (k - m) = 2m - k.$$

Note that $r(k, m)$ could be negative. If this is the case, then one needs to start with $|2m - k|$ brushes at the special leaf to try to clean the tree optimally. Another observation is that when $r(k, m) = 0$, then no optimal cleaning is possible (at least one brush has to start or finish at the special leaf). Our goal is to show that it is possible to start with an initial distribution of brushes so that the process can be almost optimal.

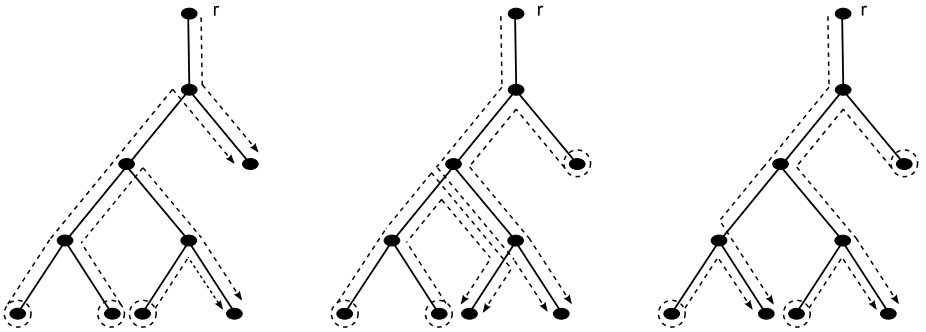

Fig. 5. The smallest configuration that is 2-optimal.

Definition 6.4. Let (T, r), $T \in \mathcal{T} \setminus \{K_1\}$, $r \in V(T)$, $\deg(r) = 1$, be a rooted tree with k ordinary leaves. Let m be such that $0 \leq m \leq k$. We say that the configuration $c(k, m)$ is δ-**optimal** (with respect to (T, r)) if there exists an initial configuration, consisting of m brushes occupying m ordinary leaves and $\max\{-(2m - k - \delta), 0\}$ brushes placed at the special leaf, from which the tree T can be cleaned so that $\max\{2m - k - \delta, 0\}$ brushes end at the special leaf. In particular, we say that $c(k, m)$ is optimal if it is 0-optimal.

As we already mentioned, there are some configurations that are not optimal (such as when $r(k, m) = 0$). It turns out that there are also some configurations that are not 1-optimal. Fig. 5 presents all non-isomorphic initial configurations of 3 brushes at ordinary leaves showing that for this tree the configuration $c(5, 3)$ is 2-optimal which means that one brush at the special leaf is needed to clean the tree. The next lemma, however, shows that every configuration is δ-optimal for some $\delta \in \{0, 1, 2\}$.

Lemma 6.5. *Let (T, r), $T \in \mathcal{T} \setminus \{K_1\}$, $r \in V(T)$ be a rooted tree with k ordinary leaves. Let m be such that $0 \leq m \leq k$. Then, the following hold.*

(i) *If $m \in \{0, k\}$, then the configuration $c(k, m)$ is optimal.*
(ii) *If $0 < m < k$ and $m \neq (k + 1)/2$, then the configuration $c(k, m)$ is δ-optimal for some $\delta \in \{0, 1\}$.*
(iii) *If $m = (k + 1)/2$, then the configuration is δ-optimal for some $\delta \in \{0, 2\}$. Moreover, if $\delta = 2$, then*
  (a) *there exists m', $0 < m' < m$, such that the configuration $c(k, m')$ is optimal.*
  (b) *there exists m'', $m < m'' < k$, such that the configuration $c(k, m'')$ is optimal.*

Note that for $T = K_2$ both possible configurations are optimal by (i) (while (iii) holds with $\delta = 0$).
One further observation which we need to make is that vertices of degree 2 in trees do not affect the brush number.

Definition 6.6. Let T be a tree on n vertices, t of which have degree 2. The **homeomorphic reduction** of T is the tree $\widehat{T} = T_t$ where $T_0 = T$ and for each $i \in \{1, 2, \ldots, t\}$, T_i is obtained from T_{i-1} by contracting an edge that has an end-vertex of degree 2 in T_{i-1}.

Note that $\widehat{T}$ has no vertices of degree 2 (because each contraction of an edge that has an end-vertex of degree 2 reduces the number of vertices of degree 2 by 1), and that T can be constructed from $\widehat{T}$ by performing a sequence of t edge subdivisions. Clearly $d_\ell(T) = d_\ell(\widehat{T})$. Moreover, it is readily observed that any cleaning strategy for T corresponds to a cleaning strategy for $\widehat{T}$, and vice-versa, and hence the validity of the following lemma is apparent:

Lemma 6.7. *If T is a tree then $B(T) = B(\widehat{T})$.*

We will now show that the main theorem is an easy consequence of the foregoing lemmata.

Proof of Theorem 6.2. It follows from Lemma 6.3 that it is enough to prove the theorem for trees from the family $\mathcal{T}$. In order to prove part (i), suppose that $T \in \mathcal{T}$ is any tree with $d_\ell(T)$ vertices of degree 1 and that $d_\ell(T)$ is an odd number. Select any vertex of degree 1 and call it the special leaf; hence the remaining vertices of degree 1 are the ordinary leaves. We apply Lemma 6.5 with $k = d_\ell(T) - 1$ (k even) and $m = k/2$ to get that $c(k, m)$ is 1-optimal. This implies that there exists an initial configuration of brushes consisting of m brushes placed on m ordinary leaves and 1 brush placed on the special leaf. It follows that $B(T) \leq m + 1 = \lceil d_\ell(T)/2 \rceil$ and, since a lower bound is obvious, this finishes the proof of part (i).

Part (ii) could be proved using Lemma 6.5 one more time but, in fact, it follows easily from part (i). Suppose that $T \in \mathcal{T}$ is any tree for which $d_\ell(T)$ is an even number. If $T = K_2$ it is apparent that the result holds. So we now assume that $T \neq K_2$ and note that each internal vertex of T has degree 3. Select any vertex v of degree 1 and observe that the tree $T' = T \setminus v$ has $d_\ell(T) - 1$ vertices of degree 1, as does $\widehat{T'}$ (which is in $\mathcal{T}$ even if T' is not). Since $d_\ell(T) - 1$ is odd, it follows from Lemma 6.7 and part (i) that

$$B(T \setminus v) = B(\widehat{T'}) = \left\lceil \frac{d_\ell(T) - 1}{2} \right\rceil = \frac{d_\ell(T)}{2},$$

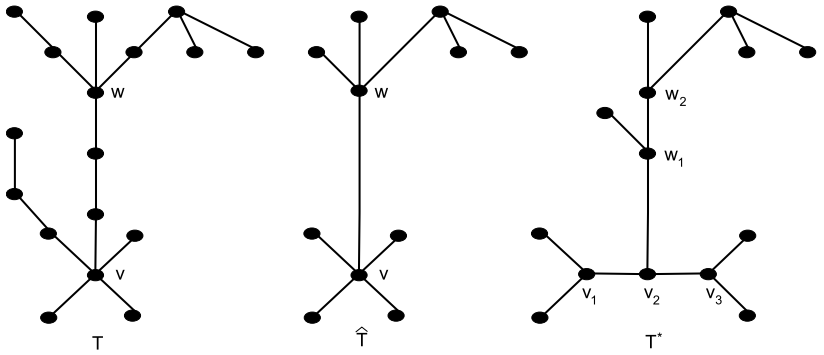

Fig. 6. $T, \widehat{T}$ and T^*.

and so $T \setminus v$ can be cleaned with $d_\ell(T)/2$ brushes. One can put an additional brush on the vertex v to clean T and so $B(T) \leq d_\ell(T)/2 + 1$. The proof is finished, since a lower bound of $d_\ell(T)/2$ is obvious. □

All that remains is to prove Lemmas 6.3 and 6.5.

6.1. Proof of Lemma 6.3

Given a tree T, we show how to construct a tree T^* such that $d_\ell(T) = d_\ell(T^*)$, $T^* \in \mathcal{T}$, and $\widehat{T}$ is a graph minor of T^*.

Definition 6.8. Given a tree T, we associate with its homeomorphic reduction $\widehat{T}$ a tree T^* on $\sum_{v \in V(\widehat{T})} f(v)$ vertices where

$$f(v) = \begin{cases} 1 & \text{if } \deg_{\widehat{T}}(v) \in \{1, 3\} \\ \deg_{\widehat{T}}(v) - 2 & \text{otherwise.} \end{cases}$$

Each vertex v of $\widehat{T}$ gives rise to a path $(v_1, v_2, \ldots, v_{f(v)})$ of order $f(v)$ in T^*. If $uv \in E(\widehat{T})$ then in T^* there will exist a single edge of the form $u_i v_j$ for one i $(1 \leq i \leq f(u))$ and one j $(1 \leq j \leq f(v))$. The $\deg_{\widehat{T}}(v)$ edges of this type that arise from the edges incident with v in $\widehat{T}$ will be distributed among $v_1, v_2, \ldots, v_{f(v)}$ so that if $\deg_{\widehat{T}}(v) \geq 4$ then v_1 and $v_{f(v)}$ are each incident with two such edges and each of $v_2, \ldots, v_{f(v)-1}$ are incident with one such edge, whereas if $\deg_{\widehat{T}}(v) \leq 3$ then all $\deg_{\widehat{T}}(v)$ of these edges will be incident with v_1 in T^*.

An example of a tree T, its homeomorphic reduction $\widehat{T}$ and an associated tree T^* are illustrated in Fig. 6. Note that different choices of distributions of edges of the form $u_i v_j$ may result in several non-isomorphic choices for T^*. However, for our purposes, any one of them will suffice.

It is apparent that $d_\ell(T) = d_\ell(\widehat{T}) = d_\ell(T^*)$.

Since $B(T) = B(\widehat{T})$, to show that $B(T) \leq B(T^*)$ it suffices to show that any cleaning strategy for T^* can be used to establish a cleaning strategy for $\widehat{T}$ with the same number of brushes. Equivalently, we show that an oriented path covering of T^* with $B(T^*)$ directed paths can be used to obtain an oriented path covering of $\widehat{T}$ with $B(T^*)$ paths. Doing so is straightforward: when a path of the covering of T^* uses vertices $v_{i_1}, v_{i_2}, \ldots, v_{i_m}$ which arose from vertex v of $\widehat{T}$, it uses them consecutively within the path and so this portion of the path in T^* can be contracted into a single vertex, which would be the vertex v of $\widehat{T}$. Performing all possible similar contractions results in each directed path of the covering in T^* becoming a directed path in a covering of $\widehat{T}$.

6.2. Proof of Lemma 6.5

Let (T, r), $T \in \mathcal{T} \setminus \{K_1\}$, $r \in V(T)$, $deg(r) = 1$, be a rooted tree with k ordinary leaves $(k \geq 1)$. Note that if $m = k$, then the tree T can be easily cleaned once every single ordinary leaf starts with one brush. Indeed, partition vertices of T with respect to their distance from the special leaf. After cleaning vertices 'layer by layer' (starting from the set of vertices that is as far from the special leaf as possible), the special leaf receives k brushes and so the configuration $c(k, k)$ is optimal. It follows from the Reversibility Theorem (Theorem 3.1) that the final configuration can be reused so that every ordinary leaf receives exactly one brush. This proves that $r(k, 0)$ is also optimal and part (i) is finished.

In order to prove parts (ii) and (iii), we will use induction on k, the number of ordinary leaves of T. There is only one graph with $k = 1$ to investigate, and it is easy to check that this graph satisfies the desired properties (note that (iii) holds with $\delta = 0$). Suppose then that the properties (ii) and (iii) hold for all trees with the number of ordinary leaves smaller than some value k and our goal is to show that they hold for a given rooted tree (T, r) with k ordinary leaves. The special leaf r is adjacent to an internal vertex v that is the special leaf of two subtrees T_1 and T_2 with k_1 and k_2 ordinary leaves, respectively. We may assume that $1 \leq k_1 \leq k_2 < k$ and the inductive hypothesis can be applied to both subtrees. The general strategy

will be to start with some configuration of brushes, clean T_1 as best as possible (using the inductive hypothesis) which results with some number of brushes arriving at v. As before, any negative value that might follow from the calculations implies that there is a deficit on v and so we need to send some brushes from v to T_1. On the other hand, a positive value implies that some brushes arrive at v after T_1 is cleaned, and these brushes can be reused to clean the remaining part of the graph. The main problem is to avoid obtaining zero, since this means that all brushes coming from T_1 are to be sent to T_2 but an extra brush needs to be introduced to clean an edge between v and r, the special leaf of T. We will continue using this notation until the end of this proof. There are a few cases to consider.

Case 1. $0 < m \leq k_1 - 1$ and $m \neq (k_1 + 1)/2$. Since $c(k_1, m)$ is δ-optimal for some $\delta \in \{0, 1\}$, we can start with m brushes distributed on the ordinary leaves of T_1, clean T_1, and arrive with $2m - k_1 - \delta$ brushes at v. Since k_2 brushes need to be sent to T_2,

$$r(k, m) = (2m - k_1 - \delta) - k_2 = 2m - k - \delta$$

brushes will arrive at the special leaf of T, provided that $r(k, m) \neq 0$. Since $m \leq k_1 - 1 \leq k/2 - 1$, we have

$$r(k, m) \leq 2(k/2 - 1) - k - \delta \leq -2 < 0,$$

so no additional brush has to be introduced. The configuration $c(k, m)$ is δ-optimal.

Case 2. $0 < m = (k_1 + 1)/2 \leq k_1 - 1$. Note that in this case $3 \leq k_1 \leq k_2$. It follows from the inductive hypothesis (part (iii)) that the configuration $c(k_1, m)$ is δ_1-optimal for some $\delta_1 \in \{0, 2\}$. If $\delta_1 = 0$, then we can distribute brushes on T_1 only, clean it, and push k_2 brushes to T_2 as before. Even if $r(k, m) = 0$, one additional brush can be added to get that $c(k, m)$ is δ-optimal for some $\delta \in \{0, 1\}$. The proof for the $\delta_1 = 2$ case is slightly more complicated. It follows from the inductive hypothesis (part (iii a)) that there exists m', $0 < m' < m$, such that the configuration $c(k_1, m')$ is optimal. We can start with m' brushes distributed on T_1 and clean T_1 optimally. The remaining $m - m'$ brushes must be distributed on T_2. Since $m - m' < m = (k_1 + 1)/2 \leq (k_2 + 1)/2$, the configuration $c(k_2, m - m')$ is δ_2 optimal for some $\delta_2 \in \{0, 1\}$. Hence, the number of brushes arriving at the special leaf of T is

$$r(k, m) = (2m' - k_1) + (2(m - m') - k_2 - \delta_2) = 2m - k - \delta_2,$$

as we wish, provided $r(k, m) \neq 0$. On the other hand,

$$r(k, m) = 2m - k - \delta_2 \leq (k_1 + 1) - k = 1 - k_2 \leq -2 < 0.$$

No brush needs to be added and so $c(k, m)$ is δ_2-optimal.

Case 3. $m = k_1$. This case is easy. We put one brush on every ordinary leaf of T_1 and send them all through vertex v to T_2, introducing some brushes on the special leaf of T_1, if needed. If $k_1 \neq k_2$, then $c(k, m)$ is optimal ($r(k, m) = k_1 - k_2 = 2m - k \neq 0$); otherwise, one additional brush is added to clean the graph and so $c(k, m)$ is 1-optimal ($r(k, m) = k_1 - k_2 - 1 = 2m - k - 1$).

Case 4. $k_1 + 1 \leq m < k$, $m \neq (k + 1)/2$, and $m - k_1 \neq (k_2 + 1)/2$. We start with k_1 brushes distributed on the ordinary leaves of T_1 and $m - k_1$ brushes on T_2. Since $m - k_1 \neq (k_2 + 1)/2$, the configuration $c(k_2, m - k_1)$ is δ-optimal for some $\delta \in \{0, 1\}$. The number of brushes arriving at v is equal to

$$k_1 + (2(m - k_1) - k_2 - \delta) = 2m - k - \delta \neq 1 - \delta,$$

since $m \neq (k + 1)/2$. This implies that no extra brush needs to be introduced if $\delta = 1$. For $\delta = 0$, it might be the case that one brush has to be added but this causes no problem. We get that the configuration $c(k, m)$ is either optimal or 1-optimal.

Case 5. $k_1 + 1 \leq m = (k + 1)/2 < k$. As before, we start with k_1 brushes distributed on the ordinary leaves of T_1 and $m - k_1$ brushes on T_2. Since

$$m - k_1 = \frac{k + 1}{2} - k_1 = \frac{k_2 + 1 - k_1}{2} \neq \frac{k_2 + 1}{2},$$

the configuration $c(k_2, m - k_1)$ is δ-optimal for some $\delta \in \{0, 1\}$. If $\delta = 0$, then $2m - k = 1$ brush arrives at vertex v and it can be pushed to the special leaf of T. The configuration $c(k, m)$ is therefore optimal. If $\delta = 1$, then one additional brush has to be added to get that $c(k, m)$ is 2-optimal (which is allowed in this case). It remains to show that there exist m', m'' such that $0 < m' < m < m'' < k$ and the corresponding configurations are optimal.

Since $m \geq k_1 + 1$, we put $m' = k_1 < m$ to get the first desired optimal configuration $c(k, m')$ starting with brushes occupying all ordinary leaves of T_1. On the other hand, we can take $m'' = k_2$. Indeed, since $(k + 1)/2 \leq k_2 + (1/2)$, we have that $m < k_2 = m''$. Then one can start with brushes occupying all ordinary leaves of T_2 to get an optimal configuration $c(k, m'')$, either as the original cleaning sequence if $m = k_2$ or as the second desired cleaning sequence with $m < m''$.

Case 6. $m - k_1 = (k_2 + 1)/2$. First note that $m = k_1 + (k_2 + 1)/2 > (k + 1)/2$ and so our task is to show that the corresponding configuration is optimal or 1-optimal. It follows from the inductive hypothesis that $c(k_2, m - k_1)$ is δ_2-optimal for some $\delta_2 \in \{0, 2\}$. If $\delta_2 = 0$, then

$$r(k, m) = k_1 + (2(m - k_1) - k_2) = k_1 + 1 \geq 2$$

brushes arrive at v and can be sent to the special leaf of T. The configuration $c(k, m)$ is optimal.

The $\delta_2 = 2$ case is more complicated. We need to consider two sub-cases. Suppose first that $k_2 \geq m$. Since $c(k_2, m)$ is δ-optimal for some $\delta \in \{0, 1\}$, we can start with m brushes on T_2 to get

$$r(k, m) = (2m - k_2 - \delta) - k_1 = k_1 + 1 - \delta \geq 1.$$

The configuration $c(k, m)$ is δ-optimal as well. Suppose now that $k_2 < m$. We might want to start with k_2 brushes on T_2 and $m - k_2$ brushes on T_1. If $c(k_1, m - k_2)$ is δ-optimal for some $\delta \in \{0, 1\}$, then the configuration $c(k, m)$ is δ-optimal as well. However, it might happen that $c(k_1, m - k_2)$ is 2-optimal. But this implies that $m - k_2 = (k_1 + 1)/2$; that is $k_1 = k_2$. In such case we need to use part (iii b) to deduce that there exists m'' such that $m - k_2 < m'' < k_1$ and that $c(k_1, m'')$ is optimal. We start with m'' brushes on T_1 and $m - m''$ on T_2 (note that $m - m'' < k_2$ so this is feasible). Since $m'' < k_1$, we have $m - m'' > m - k_1 = (k_2 + 1)/2$ and the configuration $c(k_2, m - m'')$ is δ-optimal for some $\delta \in \{0, 1\}$. As before $r(k, m) = k_1 + 1 - \delta \geq 1$ which implies that $c(k, m)$ is δ-optimal, which concludes the proof of Lemma 6.5.

7. Concluding remarks

Having established in Section 6 that trees with an even number of vertices of degree 1 can be partitioned into two sets depending on whether the brush number equals $d_\ell(T)/2$ or exceeds it by 1, it is natural to ask for a characterization of trees that belong to one of these two sets. We are also curious about the decision problem: Given a tree T for which $d_\ell(T)$ is even, is $B(T) = d_\ell(T)/2$? We leave it as an open question to determine the computational complexity of this problem.

Acknowledgements

The first author gratefully acknowledges the support of the Australian Research Council via grants DP0770400, DP120100790 and DP120103067. The fourth author gratefully acknowledges support from NSERC, CFI and IRIF. The fifth author gratefully acknowledges support from NSERC and Ryerson University.

References

[1] N. Alon, P. Prałat, N. Wormald, Cleaning regular graphs with brushes, SIAM J. Discrete Mathematics 23 (2008) 233–250.
[2] A. Bjorner, L. Lovasz, P.W. Shor, Chip-firing games on graphs, European J. Combin. 12 (1991) 283–291.
[3] K. Efe, G.-L. Feng, A proof for Bisection width of grids, World Acad. Sci. Eng. Tech. 27 (2007) 172–177.
[4] S. Gaspers, M.-E. Messinger, R.J. Nowakowski, P. Prałat, Clean the graph before you draw it!, Inform. Process. Lett. 109 (2009) 463–467.
[5] S. Gaspers, M.-E. Messinger, R.J. Nowakowski, P. Prałat, Parallel cleaning of a network with brushes, Discrete Appl. Math. 158 (2009) 467–478.
[6] P. Gordinowicz, R. Nowakowski, P. Prałat, POLISH—Let us play the cleaning game, Theoret. Comput. Sci. 463 (2012) 123–132.
[7] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Morgan Kaufmann, 1992.
[8] D. Lokshtanov, N. Misra, S. Saurabh, Imbalance is fixed parameter tractable, in: Computing and Combinatorics, in: Lecture Notes in Computer Science, vol. 6196, 2010, pp. 199–208.
[9] S.G. McKeil, Graph Cleaning, M.Sc. Thesis, Dalhousie University, 2007.
[10] M.-E. Messinger, R.J. Nowakowski, P. Prałat, Cleaning a network with brushes, Theoret. Comput. Sci. 399 (2008) 191–205.
[11] M.-E. Messinger, R.J. Nowakowski, P. Prałat, Cleaning with Brooms, Graphs Combin. 27 (2011) 251–267.
[12] M.E. Messinger, R.J. Nowakowski, P. Prałat, Elimination schemes and lattices, preprint.
[13] M.-E. Messinger, R.J. Nowakowski, P. Prałat, N. Wormald, Cleaning random d-regular graphs with brushes using a degree-greedy algorithm, in: Proceedings of the 4th Workshop on Combinatorial and Algorithmic Aspects of Networking, CAAN 2007, in: Lecture Notes in Computer Science, vol. 4852, Springer, 2007, pp. 13–26.
[14] K. Nakano, Linear layout of generalized hypercubes, Internat. J. Found Comput. Sci. 14 (2003) 137–156.
[15] C.H. Papadimitrou, M. Sideri, The bisection width of grid graphs, Theory Comput. Syst. 29 (1996) 97–110.
[16] P. Prałat, Cleaning random d-regular graphs with Brooms, Graphs Combin. 27 (2011) 567–584.
[17] P. Prałat, Cleaning random graphs with brushes, Aust. J. Combin. 43 (2009) 237–251.