Clean the graph before you draw it! [*]

Serge Gaspers[†] Margaret-Ellen Messinger[‡] Richard J. Nowakowski[‡]
Paweł Prałat[‡]

[†]Department of Informatics, University of Bergen,
N-5020 Bergen, Norway
[‡]Department of Mathematics and Statistics, Dalhousie University,
Halifax, NS, B3H 3J5, Canada

Abstract

We prove a relationship between the Cleaning problem and the Balanced Vertex–Ordering problem, namely that the minimum total imbalance of a graph equals twice the brush number of a graph. This equality has consequences for both problems. On one hand, it allows us to prove the $\mathcal{NP}$-completeness of the Cleaning problem, which was conjectured by Messinger et al. [7]. On the other hand, it also enables us to design a faster algorithm for the Balanced Vertex–Ordering problem [6].

1 Introduction

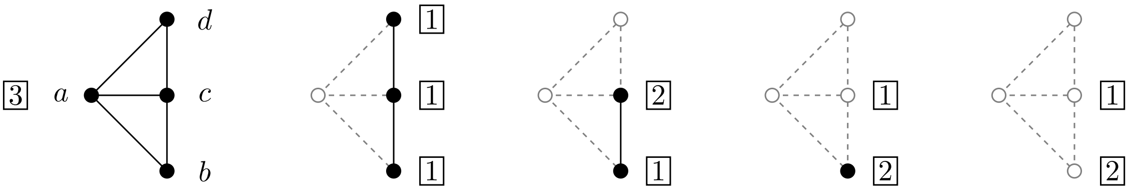

Figure 1: A graph G with an initial configuration of 3 brushes.

In Figure 1 we illustrate the Cleaning problem. All edges and vertices are initially dirty and we place three brushes at vertex a. As vertex a contains at least as many brushes as dirty incident edges, it may be cleaned. At step 1, vertex a is *cleaned* and a brush is sent down each dirty edge. At step 2, vertex c cannot be cleaned as it contains one brush, but has two dirty incident edges. However, either of b and d can be cleaned: In Figure 1, we clean d and one brush is send down the dirty edge, cleaning it. At step 3, either of c and b could be cleaned. We clean c, but since only one brush can traverse an edge, one brush remains at c. Finally, b is cleaned although no brushes move. All edges and vertices of the graph have been

[*]Partially supported by grants from the NFR, NSERC, and MITACS serge@ii.uib.no, messnger@mathstat.dal.ca, rjn@mathstat.dal.ca, pralat@mathstat.dal.ca

cleaned, using a total of three brushes. One objective is to determine the minimum number of brushes needed to clean a graph (over all possible initial distributions of brushes). The problem has been introduced in [7] where some initial results can be found as well. Cleaning random graphs was considered in [1, 8, 10] whereas the parallel version of the model has been studied in [3].

Graph drawing is the art of drawing graphs in such a way that the relationship between the objects (in the graph) are easily understood by looking at the picture. For the starting point of many graph drawing algorithms, a 'balanced' ordering of the vertices is used [4, 5, 9, 11, 12]. Although there are multiple ways to define a balanced ordering, for our purposes it implies that the neighbours of each vertex are distributed as evenly as possible. In the Balanced Vertex–Ordering problem, one wants to find a linear layout of the graph so that for each vertex, the number of incident edges drawn to the right is close to the number of incident edges drawn to the left.

A (good) heuristic for the Cleaning problem is to find a cleaning sequence where, for each vertex, as many of the incoming brushes as possible can be used when that vertex is cleaned. Obviously, the cleaning sequence can be re-interpreted as a linear layout with a constraint very similar to the Balanced Vertex–Ordering problem. In this note, we make this explicit in Theorem 4.1, but first we need to formalize the problems and introduce some results. This allows us to show that Cleaning is NP-complete (Theorem 4.3) and give an improved algorithm for solving Balanced Vertex–Ordering (Corollary 4.7).

2 Definitions

At each step t, $\omega_t(v)$ denotes the number of brushes at vertex v and D_t denotes the set of dirty vertices. An edge $uv \in E$ is dirty if and only if both u and v are dirty: $\{u, v\} \subseteq D_t$. Finally, let $D_t(v)$ denote the number of dirty edges incident to v at step t:

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

The initial configuration of brushes, denoted ω_0, is simply the starting collective distribution of brushes on a graph where $\omega_0(v)$ denotes the number of brushes initially at vertex v.

Definition 2.1. *[7] The* ***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* ($T = t$)*, return the* ***cleaning sequence*** $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_T)$*, the* ***final set of dirty vertices*** D_T*, and the* ***final configuration of brushes*** ω_T

(2) *Clean* α_{t+1} *and all dirty incident edges by traversing a brush from* α_{t+1} *to each dirty neighbour. Consequently,* $D_{t+1} = D_t \setminus \{\alpha_{t+1}\}$*,* $\omega_{t+1}(\alpha_{t+1}) = \omega_t(\alpha_{t+1}) - D_t(\alpha_{t+1})$*, and for every* $v \in N(\alpha_{t+1}) \cap D_t$*,* $\omega_{t+1}(v) = \omega_t(v) + 1$*, otherwise* $\omega_{t+1}(v) = \omega_t(v)$*.*

(3) *Set* $t := t + 1$ *and go back to* **(1)**

It was shown in [7] that given an initial configuration ω_0, the cleaning process returns a unique final set of dirty vertices. Consequently, if there exists a cleaning sequence that will clean G, we know that every (legal) cleaning sequence will clean G.

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

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

$$b(G) = \min_{\omega_0 : V \to \mathbb{N} \cup \{0\}} \Big\{ \sum_{v \in V} \omega_0(v) : G \text{ can be cleaned by } \omega_0 \Big\}.$$

Similarly, $b_\alpha(G)$ is defined as the minimum number of brushes needed to clean G using the cleaning sequence α.

The Cleaning problem is defined as follows.

Cleaning

Instance: A graph $G = (V, E)$ and integer $k \geq 0$.
Question: Is $b(G) \leq k$?

Before we introduce a problem which is related to the Cleaning one, we need a few more definitions.

Definition 2.3. *For a graph $G = (V, E)$ with $|V| = n$, given a cleaning sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, let*

$$N^-_\alpha(\alpha_{t+1}) = \{\alpha_i \in N(\alpha_{t+1}) \ : \ i < t+1\} \text{ and } N^+_\alpha(\alpha_{t+1}) = \{\alpha_i \in N(\alpha_{t+1}) \ : \ i > t+1\}.$$

A vertex–ordering π of a graph $G = (V, E)$ on n vertices is a linear layout or more precisely, a bijection $\pi : V \to \{1, 2, \ldots, n\}$. Given a graph $G = (V, E)$ and a vertex–ordering π of G, the *imbalance* of a vertex $v \in V$ with respect to π is

$$\phi_\pi(v) = \big| |N^+_\pi(v)| - |N^-_\pi(v_t)| \big|$$

and the *total imbalance* $\mathrm{Imb}_\pi(G)$ of an ordering is the sum of the imbalance of each vertex: $\mathrm{Imb}_\pi(G) = \sum_{v \in V} \phi_\pi(v)$. Let $\mathrm{Imb}(G)$ denote the *minimum total imbalance* taken over all possible vertex–orderings. A vertex v is said to be imbalanced in a vertex–ordering π if $\phi_\pi(v) > 0$.

Balanced Vertex–Ordering

Instance: A graph $G = (V, E)$ and integer $k \geq 0$.
Question: Does G have a vertex–ordering with total imbalance at most k?

Definition 2.4. *Let $\#\phi_\pi(G)$ denote the number of imbalanced vertices in the vertex–ordering π for G. Additionally, if Imb$(G) \leq k$, then let $\#\phi(G;k)$ denote the minimum number of imbalanced vertices in any vertex ordering π of total imbalance at most k; otherwise (that is, if Imb$(G) > k$) let $\#\phi(G;k) = k$.*

We define the corresponding notions for the number of vertices having at least one brush in an initial configuration of brushes.

Definition 2.5. *Let $\#b_\pi(G)$ denote the minimum number of vertices that have at least one brush in an initial configuration of brushes for G such that G can be cleaned using the cleaning sequence π. Additionally, if $b(G) \leq k$, then let $\#b(G;k)$ denote the minimum number of vertices that have at least one brush in an initial configuration of brushes for G such that G can be cleaned using any cleaning sequence π such that $b_\pi(G) \leq k$; otherwise (that is, if $b(G) > k$) let $\#b(G;k) = k - d_2 + 1$ where $(d_1, d_2, \ldots, d_{|V|})$ denotes the sequence of vertex degrees of $G = (V,E)$ where $d_i \leq d_{i+1}$ for all $1 \leq i \leq |V| - 1$.*

Let us note that if $b(G) \leq k$, then $\#b(G;k) \leq k - d_1 + 1$ since in order to start the process, the first vertex we clean must have at least d_1 brushes in the initial configuration. Moreover, since at least one vertex in the initial or final configuration must have at least d_2 brushes, and both configurations can be used to clean the graph by the Reversibility Theorem (Theorem 3.1), this can be further improved to $k - d_2 + 1$. This implies that $\#b(G;k) \leq k - d_2 + 1$ regardless whether G can be cleaned using at most k brushes.

3 Previous Results

In our proofs we need the Reversibility Theorem of Messinger et al. [7].

Theorem 3.1 ([7])**.** *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 $\tau_n = \omega_0$.*

The complexity of the Balanced Vertex–Ordering problem has been studied by Biedl et al. [2] and by Kára et al. [6]. It is clear that Balanced Vertex–Ordering is in $\mathcal{NP}$. Given a graph and an ordering of its vertices, it is straightforward to check whether or not that ordering has total imbalance at most k. Using a reduction from NAE-3Sat, it was shown in [2] that Balanced Vertex–Ordering is $\mathcal{NP}$-complete. In particular, Biedl et al. proved that given a graph G on $d_o(G)$ vertices of odd degree, the problem of deciding whether Imb$(G) \leq d_o(G)$ is $\mathcal{NP}$-complete and remains $\mathcal{NP}$-complete for bipartite graphs with maximum degree 6. A linear time algorithm, MEDIAN PLACEMENT, due to [2] provides a cleaning sequence that uses at most $|E(G)|/2 + (1/2)\lfloor |V(G)|/2 \rfloor$ brushes. It computes a vertex–ordering of minimum total imbalance for any graph of maximum degree at most 3.

In an effort to close the gap in the complexity results with respect to the maximum degree of the graph, it was shown in [6] that Balanced Vertex–Ordering is $\mathcal{NP}$-complete for planar graphs with maximum degree 4 and for 5-regular graphs. In particular, they show that it is $\mathcal{NP}$-complete to decide whether a graph G has minimum total imbalance at most

$$d_o(G) + d_1 - (d_1 \bmod 2) + d_2 - (d_2 \bmod 2) \tag{1}$$

where $(d_1, d_2, \ldots, d_{|V|})$ denotes the sequence of vertex degrees of $G = (V, E)$ where $d_i \leq d_{i+1}$ for all $1 \leq i \leq |V| - 1$, and $d_o(G)$ denotes the number of odd degree vertices. This bound holds because the following two facts are true for any ordering:

- every vertex of odd degree has imbalance at least one, and
- the two vertices at the beginning and at the end of any ordering have imbalance equal to their degrees.

It has also been shown by Kára et al. [6] that BALANCED VERTEX–ORDERING can be solved in time $O(n^{\#\phi(G;k)}(n + m))$.

4 New Results

Given a cleaning sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, it is easy to compute $b_\alpha(G)$. To do this, we can simply run the cleaning process on G using the cleaning sequence α and at the t^{th} step, we add the minimum number of brushes required to clean vertex α_t and continue the process. After α_t has been cleaned, it is clear that vertex α_{t+1} has $N^+_\alpha(\alpha_{t+1}) = D_t(\alpha_{t+1})$ dirty incident edges and $N^-_\alpha(\alpha_{t+1}) = \deg(\alpha_{t+1}) - D_t(\alpha_{t+1})$ clean incident edges. So α_{t+1} must have received exactly $\deg(\alpha_{t+1}) - D_t(\alpha_{t+1})$ brushes from neighbouring vertices. If

$$(\deg(\alpha_{t+1}) - D_t(\alpha_{t+1})) - D_t(\alpha_{t+1}) \geq 0,$$

then α_{t+1} requires no additional brushes and we may set $\omega_0(\alpha_{t+1}) = 0$. Otherwise, α_{t+1} requires an additional

$$D_t(\alpha_{t+1}) - (\deg(\alpha_{t+1}) - D_t(\alpha_{t+1})) = 2D_t(\alpha_{t+1}) - \deg(\alpha_{t+1})$$

brushes in order to be cleaned at step t so we set $\omega_0(\alpha_{t+1}) = 2D_t(\alpha_{t+1}) - \deg(\alpha_{t+1})$. Therefore, we can adjust ω_0 along the way:

$$\begin{aligned} \omega_0(\alpha_{t+1}) &= \max\{|N^+_\alpha(\alpha_{t+1})| - |N^-_\alpha(\alpha_{t+1})|, 0\} \\ &= \max\{2D_t(\alpha_{t+1}) - \deg(\alpha_{t+1}), 0\}, \end{aligned} \tag{2}$$

for $t = 0, 1, \ldots, |V| - 1$.

Similarly, as $\omega_n(\alpha_{t+1})$ represents the number of brushes at α_{t+1} in the final configuration (after every vertex has been cleaned), we have

$$\begin{aligned} \omega_n(\alpha_{t+1}) = \omega_{t+1}(\alpha_{t+1}) &= \max\{|N^-_\alpha(\alpha_{t+1})| - |N^+_\alpha(\alpha_{t+1})|, 0\} \\ &= \max\{\deg(\alpha_{t+1}) - 2D_t(\alpha_{t+1}), 0\}, \end{aligned} \tag{3}$$

for $t = 0, 1, \ldots, |V| - 1$.

Theorem 4.1. *For any graph $G = (V, E)$ and any vertex–ordering π of G, $Imb_\pi(G) = 2b_\pi(G)$.*

Proof. By the definition of Imb_π, ϕ_π, b_π and equations (2) and (3),

$$\begin{aligned}
\mathrm{Imb}_\pi(G) &= \sum_{v\in V} \phi_\pi(v) \\
&= \sum_{v\in V} \big||N_\pi^+(v)| - |N_\pi^-(v)|\big| \\
&= \sum_{v\in V} \max\{|N_\pi^+(v)| - |N_\pi^-(v)|, 0\} + \sum_{v\in V} \max\{|N_\pi^-(v)| - |N_\pi^+(v)|, 0\} \\
&= \sum_{v\in V} \omega_0(v) + \sum_{v\in V} \omega_n(v) \\
&= 2b_\pi(G).
\end{aligned}$$

□

Corollary 4.2 follows immediately from Theorem 4.1 since the brush number can be expressed as a minimum of $b_\alpha(G)$ over all vertex permutations α.

Corollary 4.2. *For any graph G, $Imb(G) = 2b(G)$.*

The following theorem follows from the complexity results in [2, 6] and Corollary 4.2. This proves a conjecture of Messinger et al. [7].

Theorem 4.3. Cleaning *is $\mathcal{NP}$-complete and remains $\mathcal{NP}$-complete for bipartite graphs of maximum degree 6, planar graphs of maximum degree 4, and 5-regular graphs.*

Moreover, it is $\mathcal{NP}$-complete to determine whether $b(G) \le d_o(G)/2$ for a graph G, even if G is restricted to the class of bipartite graphs with maximum degree 6 and it is $\mathcal{NP}$-complete to determine whether $b(G) \le (d_o(G) + d_1 - (d_1 \mod 2) + d_2 - (d_2 \mod 2))/2$ for a graph G, even if G is restricted to the class of planar graphs with maximum degree 4 or the class of 5-regular graphs.

Before we describe an algorithm for Cleaning, recall that a configuration of brushes ω_t maps each vertex $v \in V$ to the number of brushes $\omega_t(v)$ situated at that vertex at time step t. Given an initial configuration of brushes ω_0 for G, it can be checked in linear time if G can be cleaned with this initial configuration: if the cleaning process in Definition 2.1 leaves an empty set of dirty vertices, G can be cleaned with the initial configuration ω_0. Since the abritrary selection of a vertex v for which $\omega_t(v) \ge D_t(v)$ in step **(1)** does not influence the final set of dirty vertices [7], the cleaning process can easily be implemented in time $O(n+m)$ with appropriate data structures: each edge is traversed by at most one brush, and finding a vertex with at least as many brushes as incident dirty edges can be done in time $O(1)$.

As there are at most $O(n^k)$ possible initial configurations of at most k brushes, Cleaning can be solved in time $O(n^k(n+m))$. We prove a slightly stronger result.

Theorem 4.4. Cleaning *can be solved in time $O(n^{\#b(G;k)}(n+m))$.*

Proof. In the algorithm depicted in Figure 2, we assume that all the parameters passed to the function Next are passed by value, that is the value of these parameters is unchanged when the function returns. The algorithm checks whether there is a cleaning sequence such that the

```
Algorithm
  Run Step (0) of the cleaning process
  Set $k' := 0$, $\ell' := 0$ and $\omega_0(v) := 0$ for each $v \in V$
  foreach $v \in D_0$ do                    // all possibilities for the first vertex
      if Next($v, \omega_t, k', \ell', t$) then
          return true
  return false                              // no cleaning sequence using $\leq k$ brushes

Next($v, \omega_t, k', \ell', t$)
  $k' := k' + D_t(v) - w_t(v)$
  if $k' > k$ then                 // cleaning sequence requires more than $k$ brushes
      return false
  $\ell' := \ell' + 1$
  if $\ell' > \ell$ then               // need more than $\ell$ vertices with brushes initially
      return false
  $\omega_t(v) := D_t(v)$                                // add $D_t(v) - w_t(v)$ brushes to $v$
  Run Steps (1)–(3) of the cleaning process, continuing at step $t$
  if $D_t = \emptyset$ then                                 // the graph has been cleaned
      return true
  foreach $u \in D_t$ do                     // all possibilities for the next vertex
      if Next($u, \omega_t, k', \ell', t$) then
          return true
  return false
```

Figure 2: Backtracking algorithm for the Cleaning problem

corresponding initial configuration of brushes uses at most k brushes and at most ℓ vertices initially have brushes. It applies the cleaning process of Definition 2.1 as long as vertices can be cleaned. If no vertex can be cleaned any more, it goes through all possibilities of choosing a dirty vertex v as the next vertex to be cleaned and adds enough (that is, $D_t(v) - w_t(v)$) brushes to v so that it can be cleaned and the cleaning process continues if no more than k brushes have been used so far and if no more than ℓ vertices received brushes.

As the cleaning process has linear time complexity and for each of the $\leq \ell$ vertices to which the algorithm adds brushes so that the cleaning process can continue, the algorithm goes through $n - D_t \leq n$ vertices at time step t, the time complexity of the algorithm is $O(n^{\ell}(n+m))$.

To solve the Cleaning problem, run the algorithm with increasing values for ℓ, that is for $\ell \in \{0, \ldots, k - d_2 + 1\}$. As we already mentioned, if $b(G) \leq k$, then $\#b(G;k) \leq k - d_2 + 1$. Therefore, the answer to the question we consider is positive and we immediately stop the process if the algorithm returns true for some value of ℓ; otherwise (that is, when the algorithms returns false only), the answer is negative. In total the time complexity is $O(n^{\#b(G;k)}(n+m))$. □

The running time of the Cleaning problem is $O(n^{\#b(G;k)}(n+m)) = O(n^{k-d_2+1}(n+m)) = O(n^{k-1}(n+1))$ unless $d_2 = 1$. For $d_2 = 1$, we need an argument that is a little bit more sophisticated to show the following.

Corollary 4.5. CLEANING *can be solved in time* $O(n^{k-1}(n+m))$.

Proof. Assume that G is connected, otherwise run the algorithm for each connected component. If $d_2 \geq 2$, then $\#b(G;k) \leq k-1$ and the algorithm of Theorem 4.4 has running time $O(n^{k-1}(n+m))$. Otherwise, $d_2 = 1$ (we assume that G has at least 2 vertices). Let v be a vertex of degree 1 in G. In order to be cleaned, v has one brush in either the initial or final configuration of brushes corresponding to any cleaning sequence π. Suppose π is a cleaning sequence such that $\omega_0(v) = 0$. Then, by Theorem 3.1, the reverse cleaning sequence π^R of π uses the same number of brushes. So, without loss of generality, set $\omega_0(v) := 1$; that is, v has one brush in the initial configuration of brushes. It remains to call the algorithm of Theorem 4.4 with $\omega_0(v)$ initialized to 1 and k' initialized to 1. Its running time is clearly $O(n^{k-1}(n+m))$ with these initializations. □

We note in the next lemma that the imbalanced vertices of a given vertex ordering correspond to the vertices that have brushes either in the initial or final configuration of the corresponding cleaning sequence.

Lemma 4.6. *Let π be a vertex–ordering of $G = (V, E)$. Then*

$$\#b_\pi(G) \leq \lfloor \#\phi_\pi(G)/2 \rfloor , \quad \text{or} \quad \#b_{\pi^R}(G) \leq \lfloor \#\phi_\pi(G)/2 \rfloor ,$$

where π^R is π reversed.

Proof. Let ω_0^π be an initial configuration of brushes that yields $b_\pi(G)$. Since

$$\begin{aligned}
\#\phi_\pi(G) &= |\{v \in V : \phi_\pi(v) > 0\}| \\
&= |\{v \in V : ||N_\pi^+(v)| - |N_\pi^-(v)|| > 0\}| \\
&= |\{v \in V : |N_\pi^+(v)| - |N_\pi^-(v)| > 0\}| + |\{v \in V : |N_\pi^-(v)| - |N_\pi^+(v)| > 0\}| \\
&= |\{v \in V : \omega_0^\pi(v) > 0\}| + |\{v \in V : \omega_n^\pi(v) > 0\}| \\
&= |\{v \in V : \omega_0^\pi(v) > 0\}| + |\{v \in V : \omega_0^{\pi^R}(v) > 0\}| \\
&= \#b_\pi(G) + \#b_{\pi^R}(G),
\end{aligned}$$

either $\#b_\pi(G)$ or $\#b_{\pi^R}(G)$ is at most $\lfloor \#\phi_\pi(G)/2 \rfloor$. □

By Theorem 4.1 and Lemma 4.6, to each vertex–ordering with imbalance at most k and where at most ℓ vertices are imbalanced, there corresponds a cleaning sequence using at most $\lfloor k/2 \rfloor$ brushes and where at most $\lfloor \ell/2 \rfloor$ vertices initially have brushes.

Thus, we improve on the $O(n^{\#\phi(G;k)}(n+m))$ algorithm for BALANCED VERTEX–ORDERING in [6].

Corollary 4.7. BALANCED VERTEX–ORDERING *can be solved in time* $O(n^{\lfloor \#\phi(G;k)/2 \rfloor}(n+m))$.

Proof. To determine if a graph has a vertex–ordering of total imbalance at most k, run the algorithm of Theorem 4.4 to determine if it has brush number at most $k' = \lfloor k/2 \rfloor$. This algorithm has running time $O(n^{\#b(G;k')}(n+m))$. By Lemma 4.6, $\#b(G;k') \leq \lfloor \#\phi(G;k)/2 \rfloor$. □

References

[1] N. Alon, P. Prałat, and N. Wormald, Cleaning d-regular graphs with brushes, *SIAM Journal on Discrete Mathematics*, accepted, 20pp.

[2] T. Biedl, T. Chan, Y. Ganjali, M. Hajiaghayo, and D. Wood, Balanced vertex–orderings of graphs, *Discrete Applied Mathematics* **148**(1) (2005) 27–48.

[3] S. Gaspers, M.-E. Messinger, R. Nowakowski, and P. Prałat, Parallel cleaning of a network with brushes, *Discrete Applied Mathematics*, submitted, 18pp.

[4] G. Kant, Drawing planar graphs using the canonical ordering, *Algorithmica* **16** (1996) 4–32.

[5] G. Kant and X. He, Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems, *Theoretical Computer Science* **172**(1–2) (1997) 175–193.

[6] J. Kára, K. Kratochvíl, and D. Wood, On the complexity of the balanced vertex ordering problem, *Discrete Mathematics and Theoretical Computer Science* **9**(1) (2007) 193–202.

[7] M.-E. Messinger, R. J. Nowakowski, and P. Prałat, Cleaning a network with brushes, *Theoretical Computer Science* **399** (2008) 191–205.

[8] M.-E. Messinger, R. J. Nowakowski, P. Prałat, and N. Wormald, Cleaning random d-regular graphs with brushes using a degree-greedy algorithm, *Proceedings of the 4th Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN 2007)*, Lecture Notes in Computer Science **4852**, Springer, 2007, 13–26.

[9] A. Papakostas and I. G. Tollis, Algorithms for area-efficient orthogonal drawings, *Computational Geometry: Theory and Applications* **9** (1998) 83–110.

[10] P. Prałat, Cleaning random graphs with brushes, *Australasian Journal of Combinatorics*, accepted, 16pp.

[11] D. R. Wood, Minimizing the number of bends and volume in 3-dimensional orthogonal graph drawings with a diagonal vertex layout, *Algorithmica* **39** (2004) 235–253.

[12] D. R. Wood, Optimal three-dimensional orthogonal graph drawing in the general position model, *Theoretical Computer Science* **299** (2003) 151–178.