On the pseudo-achromatic number problem

Jianer Chen* Iyad A. Kanj† Jie Meng* Ge Xia‡ Fenghui Zhang*

Abstract

We study the parameterized complexity of the pseudo-achromatic number problem: Given an undirected graph and a parameter k, determine if the graph can be partitioned into at least k groups such that every two groups are connected by at least one edge. This problem has been extensively studied in graph theory and combinatorial optimization. We show that the problem has a kernel of size at most $(k-2)(k+1)$ vertices that is computable in time $O(m\sqrt{n} + k^3 m)$, where n and m are the number of vertices and edges, respectively, in the graph, and k is the parameter. This directly implies that the problem is fixed-parameter tractable. We also study generalizations of the problem and show that they are parameterized intractable.

Keywords. pseudo-achromatic number, parameterized complexity, kernel, fixed-parameter tractability

1 Introduction

The PSEUDO-ACHROMATIC NUMBER problem is to determine whether an undirected graph G can be partitioned into at least k groups/classes $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k)$ such that every two groups $\mathcal{G}_i$ and $\mathcal{G}_j$, $1 \leq i < j \leq k$, are connected by at least one edge. The problem is also referred to in the literature as the GRAPH COMPLETE PARTITION problem, and is formally defined as follows:

Definition 1 Let G be an undirected graph. We say that G has *pseudo-achromatic number* at least k if there exists a surjective function $f : V(G) \rightarrow \{1, \ldots, k\}$, such that: for all $i \neq j$ satisfying $1 \leq i, j \leq k$, there exists $u \in f^{-1}(i),\ v \in f^{-1}(j)$ such that $(u, v) \in E(G)$, where $f^{-1}(q)$ denotes the preimage of q under f.

The PSEUDO-ACHROMATIC NUMBER problem is:

> PSEUDO-ACHROMATIC NUMBER: Given an undirected graph G and a positive integer k, determine whether G has pseudo-achromatic number at least k.

We will be using the informal definition more frequently than the formal one.

It is easy to see that the PSEUDO-ACHROMATIC NUMBER problem is a generalization of the graph coloring problem (or the achromatic number problem), the latter problem requiring the groups in the partition to be independent sets.

The PSEUDO-ACHROMATIC NUMBER problem was first introduced by Gupta in 1969 [10], and since then it has been studied extensively [1, 2, 3, 4, 8, 12, 13]. The problem is known to be NP-complete even on restricted classes of graphs [3, 8, 12].

Kortsarz et al. [12] studied the approximability of the PSEUDO-ACHROMATIC NUMBER problem. It was proved in [12] that the problem has a randomized polynomial-time approximation algorithm of ratio

*Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA. Email: {chen,jmeng,fhzhang}@cs.tamu.edu.

†School of CTI, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604-2301, USA. Email: ikanj@cs.depaul.edu.

‡Department of Computer Science, Lafayette College, Easton, PA 18042, USA. gexia@cs.lafayette.edu.

$O(\sqrt{\lg n})$, which can be de-randomized in polynomial time. This upper bound on the approximation ratio was shown to be asymptotically tight under the randomized model.

The PSEUDO-ACHROMATIC NUMBER problem was also considered from the extremal graph theoretic point of view on special classes of graphs [2, 4, 13, 14, 15]. Balsubramanian et al. [1] gave a complete characterization of when the pseudo-achromatic number of the join of two graphs is the sum of the pseudo-achromatic number of the two graphs.

In the current paper we study the parameterized complexity of the PSEUDO-ACHROMATIC NUMBER problem. We show that the problem has a kernel of size at most $(k-2)(k+1)$ vertices that is computable in time $O(m\sqrt{n}+k^3m)$, where n and m are the number of vertices and edges, respectively, in the graph. This kernelization result directly gives an algorithm for the PSEUDO-ACHROMATIC NUMBER running in time $O(k^{(k-2)(k+1)} \cdot m\sqrt{n})$, thus showing that the problem is fixed-parameter tractable. The upper bound on the kernel size is obtained by developing elegant and highly non-trivial structural results, that are of independent interest.

We also study generalizations of the of the PSEUDO-ACHROMATIC NUMBER problem and prove that they are parameterized intractable. In particular, we consider the VERTEX GROUPING problem, in which an input instance has the form (G, H, k), where G and H are two graphs, and $k = |V(H)|$. The problem asks for the existence of a surjective function $f : V(G) \longrightarrow V(H)$ satisfying the property that $\forall u, v \in V(H), (u, v) \in E(H)$ if and only if $\exists x \in f^{-1}(u), y \in f^{-1}(v)$ such that $(x, y) \in E(G)$. The PSEUDO-ACHROMATIC NUMBER problem is a special case of the VERTEX GROUPING problem in which the graph H is the complete graph on k vertices. The VERTEX GROUPING problem falls into the category of clustering problems, where a clustering of the graph G into $|V(H)|$ clusters is sought such that the inter-cluster properties are imposed by the graph H. We prove some (parameterized) intractability results for the VERTEX GROUPING problem. For example, we show that the problem is W[1]-hard, even when the graph H is the h-star graph (i.e., $K_{1,h-1}$).

2 Preliminaries

The reader is referred to Downey and Fellows' book [7] for more details about parameterized complexity theory.

A *parameterized problem* is a set of instances of the form (x, k), where $x \in \Sigma^*$ for a finite alphabet set Σ, and k is a non-negative integer called the *parameter*. A parameterized problem Q is *fixed parameter tractable*, or simply FPT, if there exists an algorithm A that on input (x, k) decides if (x, k) is a yes-instance of Q in time $f(k)n^{O(1)}$, where f is a recursive function independent of $n = |x|$. In analogy to the polynomial time hierarchy, a hierarchy for parameterized complexity, called the *W-hierarchy*, has been defined. At the 0th level of this hierarchy lies the class of fixed-parameter tractable problems FPT. The class of all problems at the ith level of the W-hierarchy ($i > 0$) is denoted by *W[i]*. A parameterized-complexity preserving reduction (FPT-reduction) has been defined as follows. A parameterized problem Q is *FPT-reducible* to a parameterized problem Q' if there exists an algorithm of running time $f(k)|x|^c$ that on an instance (x, k) of Q produces an instance $(x', g(k))$ of Q' such that (x, k) is a yes-instance of Q if and only if $(x', g(k))$ is a yes-instance of Q', where the functions f and g depend only on k, and c is a constant. A parameterized problem Q is *W[i]-hard* if every problem in W[i] is FPT-reducible to Q. Many well-known problems have been proved to be $W[1]$-hard including: CLIQUE, INDEPENDENT SET, SET PACKING, DOMINATING SET, HITTING SET and SET COVER. The *parameterized complexity hypothesis*, which is a working hypothesis for parameterized complexity theory, states that W[i] $\neq$ FPT for every $i > 0$.

The notion of the fixed-parameter tractability of a problem turns out to be closely related to the notion of the problem having a good data reduction (or preprocessing) algorithm. Formally speaking, a parameterized problem Q is kernelizable if and only if there exists a polynomial-time computable reduction that maps an instance (x, k) of Q to another instance (x', k') of Q such that: (1) $|x'| \leq g(k)$ for some recursive function g, (2) $k' \leq k$, and (3) (x, k) is a yes-instance of Q if and only if (x', k') is a yes-instance of Q. The instance x' is called the *kernel* of x. It was shown that a parameterized problem is fixed-parameter tractable if and only if it admits a kernelization [9].

For a graph G we denote by $V(G)$ and $E(G)$ the set of vertices and edges of G, respectively. A *matching* M in a graph G is a set of edges such that no two edges in M share an endpoint. A matching M of G is said to *maximum* if the cardinality of M is maximum over all matchings in G. For a vertex v and a set of vertices Γ in G, we say that v is *connected to* Γ if v is adjacent to some vertex in Γ. Similarly, for two sets of vertices Γ and Γ' in G, we say that Γ is *connected to* Γ' if there exists a vertex in Γ that is connected to Γ'. For a vertex $v \in G$ we denote by $N(v)$ the set of neighbors of v in G. For a set of vertices Γ in G we denote by $N(\Gamma)$ the set of neighbors of all the vertices of Γ in G, i.e., $N(\Gamma) = \bigcup_{v \in \Gamma} N(v)$. We denote by S_h the $(h+1)$-star graph (i.e., $K_{1,h}$). The vertex of degree h in S_h is referred to as the *root* of the star, and the other h vertices are referred to as the *leaves* of the star. The *size* of the star S_h is the number of vertices in it, which is $h+1$. We say that a graph G contains S_h if S_h is a subgraph (not necessarily induced) of G.

For a background on network flows we refer the reader to [6], or to any standard book on combinatorial optimization.

3 The kernel

In this section we show how to construct a kernel of size (number of vertices) at most $(k-2)(k+1)$ for the parameterized PSEUDO-ACHROMATIC NUMBER problem. We start by presenting some structural results that are essential for the kernelization algorithm, and that are of independent interest on their own.

3.1 Structural results

The following lemma ascertain that graphs with large matchings have large pseudo-achromatic number.

Lemma 3.1 *If a graph G contains a matching of size at least $(k-1)k/2$, then the instance (G, k) is a yes-instance of the* PSEUDO-ACHROMATIC NUMBER *problem.*

PROOF. Assuming that G contains a matching of at least $(k-1)k/2$ edges, we show how to group the vertices of G into k groups $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k)$ so that every pair of groups is connected by at least one edge. For every pair of groups $(\mathcal{G}_i, \mathcal{G}_j)$ where $1 \leq i < j \leq k$, we use a distinct edge (u, v) of the matching to connect this group by mapping the vertex u to $\mathcal{G}_i$ and v to $\mathcal{G}_i$. The remaining vertices of G are mapped arbitrarily to the groups. Since there are exactly $(k-1)k/2$ pairs of groups and at least $(k-1)k/2$ edges in the matching, every pair of groups is connected under this mapping. It follows that (G, k) is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem. □

Lemma 3.2 *If a graph G contains a set of $k-1$ (mutually) vertex-disjoint stars of sizes $2, \ldots, k$, respectively, then the instance (G, k) is a yes-instance of the* PSEUDO-ACHROMATIC NUMBER *problem.*

PROOF. Let $\mathcal{S} = \{s_1, \ldots, s_{k-1}\}$ be a set of vertex-disjoint stars in G, where s_i is a copy of the star graph S_i. We will map the vertices in $\mathcal{S}$ to k groups $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k)$ such that every pair of groups is connected by at least one edge.

For $i = 1, \ldots, k-1$, we map the root of s_i to group $\mathcal{G}_{i+1}$, and we map its leaves, in a one-to-one fashion, to groups $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_i)$. The remaining vertices in G are mapped arbitrarily to the groups. Since there is no overlap between the vertices of any two stars in S, this mapping is well defined. It is very easy to verify now that every two distinct groups in $(\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k)$ are connected under the defined mapping. It follows that (G, k) is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem. □

Lemma 3.3 *If a graph G contains a collection of (mutually) vertex-disjoint stars each of size at least 2 and at most $k+1$, and such that the total number of vertices in all the stars is more than $(k-2)(k+1)$, then the instance (G,k) is a yes-instance of the* PSEUDO-ACHROMATIC NUMBER *problem.*

PROOF. Suppose that G contains a collection $\mathcal{P}$ of vertex-disjoint stars, each containing at least two vertices and at most $k+1$ vertices, and such that the total number of vertices of the stars in $\mathcal{P}$ is more than $(k-2)(k+1)$. Assume, to get a contradiction, that (G,k) is a no-instance of the PSEUDO-ACHROMATIC NUMBER problem.

Let s be a copy of the star graph S_h and s' be a copy of S'_h such that s and s' are vertex-disjoint. By *merging* s and s' we mean creating a copy of $S_{h+h'}$. Note that the size of the merged star is 1 less than the size of s plus the size of s'.

We construct from $\mathcal{P}$ a sequence of vertex-disjoint stars $\mathcal{S} = \langle s_{k-1}, \ldots, s_r \rangle$, such that s_i has size at least $i+1$, for $r \leq i \leq k-1$. The procedure that constructs these stars is as follows.

For $i = k-1$ down to 1 do: if the largest star in $\mathcal{P}$ is an S_j, where $j \geq i$, assign it to s_i, and remove it from $\mathcal{P}$; Otherwise, recursively merge the two stars of largest size in $\mathcal{P}$ until either there is only one star left in $\mathcal{P}$, and in which case the procedure halts, or the largest star in $\mathcal{P}$ is an S_j, where $j \geq i$, and in which case we assign it to s_i, remove it from $\mathcal{P}$, and proceed to the next value of i in the for loop.

If a star s_i in $\mathcal{S}$ was created without merging stars in $\mathcal{P}$, we call s_i a *single star*, otherwise, we call s_i a *merged star*.

Note the following: if s_i is a merged star created from merging a collection of stars, and if s_i is used to produce a valid grouping of G, then clearly the stars that s_i was merged from can replace s_i to produce a valid grouping of G. Therefore, assuming that (G,k) is a no-instance of the PSEUDO-ACHROMATIC NUMBER problem, the last star s_r constructed by the above procedure before halting must satisfy $r \geq 2$. Otherwise, the sequence $\mathcal{S}$ would contain a set of $k-1$ vertex-disjoint stars of sizes $2, \ldots, k$, and by Lemma 3.2, the instance (G,k) would be a yes-instance of the problem, contradicting our assumption.

Now assume that the above procedure halts after constructing a sequence of vertex-disjoint stars $\mathcal{S} = \langle s_{k-1}, \ldots, s_r \rangle$, such that s_i has size at least $i+1$, for $2 \leq r \leq i \leq k-1$.

We define a *monotone subsequence* of $\mathcal{S}$ to be a consecutive subsequence $\langle s_i, s_{i-1} \ldots, s_j \rangle$ of $\mathcal{S}$ such that either $s_i, s_{i-1} \ldots, s_j$ are all single stars, or they are all merged stars. A monotone subsequence $\langle s_i, s_{i-1} \ldots, s_j \rangle$ of $\mathcal{S}$ is *maximal* if it is maximal under containment.

Let $\langle s_i, s_{i-1} \ldots, s_{i-\ell+1} \rangle$, $\ell \geq 1$, be a maximal monotone subsequence of $\mathcal{S}$, and note that $i-\ell+1 \geq 2$ (since $r \geq 2$). We will show that the total number of vertices in the stars of $\mathcal{P}$ that were used to form the subsequence $\langle s_i, s_{i-1} \ldots, s_{i-\ell+1} \rangle$ is at most $2(i + (i-1) + \ldots + (i-\ell+1))$. We distinguish two cases:

- **Case 1.** $\langle s_i, s_{i-1}, \ldots, s_{i-\ell+1} \rangle$ consists of single stars. We distinguish two subcases:
  - **Subcase 1.1.** $i = k-1$. Since every single star contains at most $k+1$ vertices by the statement of the lemma, the total number of vertices in the stars in the subsequence is bounded by $\ell(k+1) \leq 2(k-1+k-2+\ldots+k-\ell)$. The last inequality is true because $((k-1)-\ell+1) \geq 2$.
  - **Subcase 1.2.** $i < k-1$. By the maximality of the subsequence, s_{i+1} is a merged star. Since s_i is a single star, it is easy to verify that s_i has size exactly $i+1$. The total number of vertices in the stars in the subsequence is bounded by $\ell(i+1) \leq 2(i+i-1+\ldots+i-\ell+1)$ because $i-\ell+1 \geq 2$.

- **Case 2.** $\langle s_i, s_{i-1}, \ldots, s_{i-\ell+1} \rangle$ consists of merged stars. Let s_j be any star in this subsequence, and suppose that s_j was constructed by merging stars $t_1, \ldots, t_q$ in $\mathcal{P}$. By the construction of s_j, the total number of leaves in the stars $t_1, \ldots, t_{q-1}$ is less than j (otherwise these starts would be sufficient to produce s_j), and the size of t_q is the smallest among $t_1, \ldots, t_{q-1}$. Therefore, we have:

$$|t_1| - 1 + |t_2| - 1 + \ldots + |t_{q-1}| - 1 \leq j - 1, \tag{1}$$

and

$$|t_q| \leq (|t_1| + |t_2| + \ldots + |t_{q-1}|)/(q-1). \tag{2}$$

Combining Inequality (1) with Inequality (2), and noting that $q \leq j$, we obtain:

$$|t_1| + |t_2| + \ldots + |t_q| \leq 2j. \tag{3}$$

Inequality (3) shows that the total number of vertices in the stars of $\mathcal{P}$ forming s_j is at most $2j$. By applying this inequality to each star s_j in the maximal monotone subsequence $\langle s_i, s_{i-1}, \ldots, s_{i-\ell+1} \rangle$ of merged stars, and by the linearity of addition, we obtain that the total number of vertices of $\mathcal{P}$ used to form the stars in $\langle s_i, s_{i-1}, \ldots, s_{i-\ell+1} \rangle$ is at most $2(i + (i-1) + \ldots + (i - \ell + 1))$.

It follows from the above that, for any maximal monotone subsequence $\langle s_i, s_{i-1}, \ldots, s_{i-\ell+1} \rangle$ of $\mathcal{S}$, the total number of vertices of $\mathcal{P}$ used to form the stars in this subsequence is at most $2(i+(i-1)+\ldots+(i-\ell+1))$. Applying the above bound to every maximal monotone subsequence of $\mathcal{S}$, and by the linearity of addition, we conclude that the total number of vertices in $\mathcal{P}$ forming all the stars in $\mathcal{S}$ is at most $(k-r)(k+r-1)$.

Noting that the number of remaining non-empty stars in $\mathcal{P}$ cannot form an s_{r-1}, $\mathcal{P}$ has the maximum number of vertices when $r = 2$, and there are no remaining stars in $\mathcal{P}$ to form s_1. It follows that the total number of vertices in $\mathcal{P}$ is at most $(k-2)(k+1)$, contradicting the hypothesis of the lemma.

This completes the proof. □

3.2 The decomposition of G and the auxiliary flow network J'

Let M be a maximum matching in G. Let $I = V(G) \setminus V(M)$, and note that I is an independent set in G. We partition M into three sets M_0, M_1 and M_2 as follows: M_2 is the set of edges $(u, v) \in M$ such that both u and v are connected to I; M_1 is the set of edges $(u, v) \in M$ such that exactly one vertex in the set $\{u, v\}$ is connected to I; and M_0 contains all the remaining edges in M—these are the edges $(u, v) \in M$ such that neither u nor v is connected to I.

For a vertex $u \in V(M)$ we denote by $N_I(u)$ the set $N(u) \cap I$. We denote by $N_I(M_2)$ the set $N(V(M_2)) \cap I$. We have the following lemma:

Lemma 3.4 *Let (u, v) be an edge in M_2. Then $N_I(u) = N_I(v) = \{w\}$.*

Proof. By definition, each of $N_I(u)$ and $N_I(v)$ is nonempty. Let $w_1 \in N_I(u)$ and $w_2 \in N_I(v)$. It suffices to show that $w_1 = w_2$. If not, then the path (w_1, u, v, w_2) would be an augmenting path with respect to M in G, contradicting the maximality of M. □

Let $D = I \setminus N_I(M_2)$. We partition the set $V(M_1)$ into two sets L and R such that R is the set of vertices in $V(M_1)$ that are connected to D, and L is the set of remaining vertices in $V(M_1)$. Note that, by the definition of M_1, for every edge $(u, v) \in M_1$, exactly one vertex in $\{u, v\}$ is in R and the other vertex is in L.

Let J be subgraph of G whose vertex-set is $R \cup D$ and whose edge-set is $\{(u, v) \mid u \in R \text{ and } v \in D\}$. We construct the flow network J' as follows. Add a source s and a sink t to J. For every vertex $u \in R$ add a directed edge (s, u) of capacity $k-1$ units. Direct every edge (u, v) between a vertex $u \in R$ and a vertex $v \in D$ from u to v, and assign it a capacity of 1 unit. For every vertex v in D, add a directed edge (v, t) of capacity 1 unit. This completes the construction of J'.

Let f^* be an integer-valued maximum flow in J'. Define the *flow through* a vertex u, denoted by f^*_u, to be the total outgoing flow from vertex u, i.e., $\Sigma_{(u,w)\in E(J')} f^*(u,w)$. Call a vertex $u \in R$ *saturated* if the value of f^*_u is $k-1$. For simplicity, if an edge (resp. vertex) in $J' - \{s,t\}$ is saturated, then we say that its corresponding edge (resp. vertex) in G is saturated. With this in mind, we will identify the vertices and edges in $J' - \{s,t\}$ with their counterparts in G, and we will be working on the graph G.

Let $T = \{v_1, \ldots, v_\ell\}$ be the set of vertices in D such that $f^*_{v_i} = 0$, for $i = 1, \ldots, \ell$, and let $T' = D \setminus T$. We will show next that the set of vertices T can be removed from the graph G without affecting the pseudo-achromatic number of G.

Define the sequence of subgraphs G_i, $i = 0, \ldots, \ell$, inductively as follows: $G_0 = G$, and $G_i = G_{i-1} - v_i$, for $i = 1, \ldots, \ell$. We also define the corresponding sequence of flow networks: $J'_0 = J'$, and $J'_i = J'_{i-1} - v_i$, for $i = 1, \ldots, \ell$. Note that since $f^*_{v_i} = 0$, f^* is a maximum flow in J'_i, for $i = 0, \ldots, \ell$. We have the following key lemma:

Lemma 3.5 *For $i = 1, \ldots, \ell$, if the pseudo-achromatic number of G_{i-1} is at least k then the pseudo-achromatic number of G_i is at least k.*

PROOF. Assuming that the pseudo-achromatic number of G_{i-1} is at least k, there exists a grouping $\mathcal{H}$ of G_{i-1} that maps the vertices of G_{i-1} into k groups such that each pair of groups is connected. For a vertex $v \in G_{i-1}$, denote by $\mathcal{G}(v)$ the group that contains v in $\mathcal{H}$. We call an edge e in G_{i-1} a *critical edge* (with respect to $\mathcal{H}$) if there exists a pair of groups in $\mathcal{H}$ such that e is the only edge between the two groups in this pair; otherwise, e is called a *noncritical edge*. We call a vertex v in G_{i-1} a *critical vertex* (with respect to $\mathcal{H}$) if there exists a critical edge incident on v; otherwise, v is called a *noncritical vertex*. For any vertex $v \in G_{i-1}$, there are at most $k-1$ critical edges incident on v, otherwise, by the Pigeon Hole principle, there would be at least two critical edges between some two groups in $\mathcal{H}$.

Since $G_i = G_{i-1} - v_i$, if v_i is a noncritical vertex in G_{i-1}, then the grouping $\mathcal{H}$ of G_{i-1} is also a grouping of G_i, and we are done. Suppose now that v_i is a critical vertex in G_{i-1}. It suffices to show that there exists a grouping $\mathcal{H}'$ with respect to which v_i is a noncritical vertex. The rest of the proof is dedicated to proving the previous statement.

Consider a partitioning of the set of vertices R in $V(M_1)$ (recall that R is the set of vertices in $V(M_1)$ that are connected to D) into (R_1, R_2) such that R_1 is the set of saturated vertices in R, and $R_2 = R \setminus R_1$.

For a vertex $u \in R_1$, since u is incident on exactly $k-1$ saturated edges in G_{i-1}, and since there are at most $k-1$ critical edges incident on u, the number of critical unsaturated edges incident on u is less than or equal to the number of noncritical saturated edges incident on u. Therefore, we can define a one-to-one function Φ that, for every critical unsaturated edge e incident on u, associates with e a noncritical saturated edge $\Phi(e)$.

We call a path $P : (v_i = w_0, \ldots, w_{2s})$ from v_i in G_{i-1} a *strongly alternating* path if (w_{2j}, w_{2j+1}) is a critical unsaturated edge and (w_{2j+1}, w_{2j+2}) is a noncritical saturated edge, for $j = 0, \ldots, s-1$. Note that the vertices w_{2j}, $j = 0, \ldots, s$, on a strongly alternating path belong to D, and the vertices w_{2j+1}, for $j = 0, \ldots, s-1$, belong to R. Note also that any vertex w_{2j}, $j = 0, \ldots, s-1$, on a strongly alternating path is only connected to R_1, otherwise, there exists a vertex $u \in R_2$ such that the subpath of P from v_i to w_{2j}, followed by the edge (w_{2j}, u), is a flow augmenting path with respect to f^*, contradicting the maximality of f^*.

We call a critical vertex $v \in D$ *important* if there exists a strongly alternating path that ends at v. By definition, the statement that v_i is an important vertex is vacuously true. Note that any critical edge which is incident on an important vertex v is unsaturated because there exists a noncritical saturated edge incident to v, on the strongly alternating path to v.

Let v be an important vertex. Define a recursive process **Explore-Base**(v) starting at $v = v_i$ as follows. Let $(u_1, v), \ldots, (u_r, v)$ be the critical edges incident on v, and note that from the above discussion, the edges (u_j, v), $j = 1, \ldots, r$, are unsaturated, and that the vertices $u_1, \ldots, u_r \in R_1$. For every vertex u_j,

$j = 1, \ldots, r$, there exists a vertex $d_j \in D$ such that $(u_j, d_j) = \Phi(u_j, v)$. Since every vertex in D has only one outgoing edge of capacity 1 to the sink t, the vertices d_j, $j = 1, \ldots, r$, are all distinct; otherwise, there would exist a vertex in $\{d_1, \ldots, d_r\}$ with an incoming flow of value greater than 1, contradicting the flow properties. We call the set $\{d_1, \ldots, d_r\}$ the *base* of v and denote it by B_v. If every vertex in the base of v is not critical, then the process **Explore-Base(v)** halts. Otherwise, for every critical vertex d_j in the base of v, the strongly alternating path to v followed by the path (v, u_j, d_j) is a strongly alternating path that ends at d_j, and hence, d_j is an important vertex; the process **Explore-Base** is then applied to d_j.

We argue that if B_p and B_q are two bases for two important vertices p and q, then B_p and B_q must be disjoint. Suppose not, and let w be the first vertex, with respect to the procedure **Explore-Base**, to appear in two distinct bases B_v and $B_{v'}$. Since $w \in D$, the incoming flow to w is at most 1. By definition, any strongly alternating path to w must end in a saturated edge. Therefore, if w belongs to two different bases, then the parents of w on the two strongly alternating paths corresponding to the two bases must be the same. Since the function Φ is one to one, the two grandparents of w, v and v', must be identical, and hence $B_v = B_{v'}$, contradicting the fact that B_v and $B_{v'}$ are distinct.

According to the above discussion, the process **Explore-Base(v_i)** must eventually halt.

Note that if a vertex d_j in B_v is not critical, we can modify the grouping $\mathcal{H}$ so that vertex d_j is mapped to $\mathcal{G}(v)$, and the edge (v, u_j) is no longer critical. Therefore, if every vertex in the base of v is noncritical, then there exists another grouping where v is noncritical with respect to that grouping.

Let $\mathcal{T}$ be an auxiliary rooted tree defined as follows. The root of $\mathcal{T}$ is v_i, and for a node v in $\mathcal{T}$, the children of v in $\mathcal{T}$ are the vertices in B_v. Then the leaves of $\mathcal{T}$ are noncritical vertices in G_{i-1}. Let v a non-leaf node in $\mathcal{T}$ whose children are all noncritical vertices. Then, by the above discussion, we can modify the grouping $\mathcal{H}$ so that v is a noncritical vertex. Applying this operation to $\mathcal{T}$ in a bottom-up fashion starting at the leaves of $\mathcal{T}$, the grouping $\mathcal{H}$ can be modified to obtain another grouping with respect to which the vertex v_i is a noncritical vertex. This completes the proof. □

The above lemma shows that the set of vertices T can be safely removed from the graph G.

Lemma 3.6 *Let $G' = G - T$, where T is the set of vertices defined above. Then $V(G')$ can be decomposed into a collection $\mathcal{P}$ of vertex-disjoint stars, each star of size at least 2 and at most $k + 1$.*

Proof. We will exhibit the collection of vertex-disjoint stars $\mathcal{P}$ in G'. We will denote by $V_{\mathcal{P}}$ the set of vertices of the stars in the collection $\mathcal{P}$, and by $E_{\mathcal{P}}$ the set of edges of the stars in $\mathcal{P}$.

Note that since G' is a subgraph of G, the decomposition of G described before induces a decomposition of G'. In particular, the set of vertices of G' consists of the vertices in the matching M, the vertices in $N_I(M_2)$, and the vertices in D with a non-zero flow value. For a vertex u in R, let $S(u)$ be the star graph formed by the incident edge to u in M_1, together with the set of saturated edges in G' incident on u. Clearly, each such star $S(u)$ has size at least 2 and at most $k + 1$ since the capacity of u in J' is $k - 1$. Moreover, for any two vertices u and v in R, the two star graphs $S(u)$ and $S(v)$ share no vertices; otherwise, there would be a shared vertex $w \in S(u) \cap S(v)$ of capacity 1 in J' with two saturated edges incident on it, contradicting the flow properties. We add all such stars $S(u)$ to the collection $\mathcal{P}$.

We also include in $\mathcal{P}$ a maximal set of disjoint S_2 stars such that the root of each S_2 star is a vertex in $N_I(M_2)$ and its leaves are the end points of the same edge in M_2. Moreover, for every edge in M_2 whose endpoints are not yet in $V_{\mathcal{P}}$, we include it in $\mathcal{P}$ as an S_1 stars. Finally we include in $\mathcal{P}$ the matching edges in M_0 as S_1 stars.

It is clear that all the stars included in $\mathcal{P}$ are vertex-disjoint, and that each star has size at least 2 and at most $k + 1$.

We claim that $V_{\mathcal{P}}$ contains all the vertices of G'. To see why, first observe that $E_{\mathcal{P}}$ contains all the edges in M, and hence $V_{\mathcal{P}}$ contains their endpoints. Second, since every vertex v in T' is incident on a saturated edge in G', v is included in $\mathcal{P}$. Moreover, since by definition every vertex $u \in N_I(M_2)$ forms an S_2 star with an edge (w, v) in M_2, and since by Lemma 3.4 no other vertex in $N_I(M_2)$ can form a star with the edge (w, v), it follows from the construction of $\mathcal{P}$ that $u \in V_{\mathcal{P}}$. Therefore, every vertex u in $N_I(M_2)$ is in $\mathcal{P}$, and V_P contains all the vertices of G' as desired. □

3.3 Putting it all together: the kernelization algorithm

Consider the decomposition of G defined in Subsection 3.2, and let M and T be as defined in Subsection 3.2. The kernelization algorithm is given in Figure 1.

Algorithm **PseudoAchromaticNumberKernel**

INPUT: *(G, k)*
OUTPUT: *(G′, k′)*

1. compute a maximum matching M and the set of vertices T as described in Subsection 3.2;
2. **if** $|M| \geq (k-1)k/2$ **then return** YES;
3. $G' = G - T$;
4. **if** $|V(G')| > (k-2)(k+1)$ **then return** YES;
5. **return** $(G', k' = k)$;

Figure 1: The kernelization algorithm.

Theorem 3.7 *Given an instance (G, k) of the* PSEUDO-ACHROMATIC NUMBER *problem, the algorithm* **PseudoAchromaticNumberKernel** *either decides the instance (G, k) correctly, or returns an instance (G', k') of the problem such that G' is a subgraph of G, $k' \leq k$, and (G, k) is a yes-instance if and only if (G', k') is. Moreover, the algorithm runs in time $O(m\sqrt{n} + mk^3)$, where n and m are the number of vertices and edges, respectively, in G.*

PROOF. If the size of the maximum matching M in G is at least $(k-1)k/2$, then by Lemma 3.1, G is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem. Therefore, the algorithm **PseudoAchromaticNumberKernel** makes the right decision in step 2.

If the subgraph $G' = G - T$ has a valid grouping into k groups such that every pair of groups is connected, then obviously so does G being a supergraph of G'. Conversely, if G has a valid grouping into k groups, then by Lemma 3.5, so does G'. It follows that (G, k) is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem if and only if (G', k') is.

It suffices to argue that if $|V(G')| > (k-2)(k+1)$ then G', and hence G, is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem, and the algorithm makes the right decision in step 4.

By Lemma 3.6, the set $V(G')$ can be decomposed into a collection of vertex-disjoint stars $\mathcal{P}$, each star of size at least 2 and at most $k+1$. Since $|V(G')| > (k-2)(k+1)$, it follows that the number of vertices in $\mathcal{P}$ is more than $(k-2)(k+1)$. Consequently, $\mathcal{P}$ satisfies the statement of Lemma 3.3, and G is a yes-instance of the PSEUDO-ACHROMATIC NUMBER problem.

Finally, to see that the algorithm **PseudoAchromaticNumberKernel** runs in time $O(m\sqrt{n}+mk^3)$, note first that the maximum matching M can be computed in $O(m\sqrt{n})$ time by a standard maximum matching algorithm [6]. Noting that the set R is a subset of M, and hence, has size $O(k^2)$ (otherwise the algorithm would have returned YES in step 1), and that each vertex in R has capacity $k-1$, it follows that the value of the maximum flow $|f^*|$ in J' is $O(k^3)$. Consequently, the maximum flow f^* in J' can be computed in time $O(m|f^*|) = O(mk^3)$ using the **Ford-Fulkerson algorithm** [6]. All other steps can be performed in time $O(m)$, and the theorem follows. □

Corollary 3.8 *The* PSEUDO-ACHROMATIC NUMBER *problem has a kernel of at most $(k-2)(k+1)$ vertices that is computable in time $O(m\sqrt{n}+mk^3)$, where n and m are the number of vertices and edges, respectively, in the graph, and k is the parameter.*

Remark 3.9 *Note that our upper-bound analysis of the size of the kernel returned by the algorithm* **PseudoAchromaticNumberKernel** *is tight. This can be seen by considering a graph G that consists of $(k-1)k-2=(k-2)(k+1)$ vertices which are the endpoints of $(k-1)k/2-1$ edges in a matching. The algorithm* **PseudoAchromaticNumberKernel** *on input (G,k) will return (G,k) as is, and without any modifications. Clearly, (G,k) is a no-instance of the* PSEUDO-ACHROMATIC NUMBER *problem.*

Using the $(k-2)(k+1)$ upper bound on the kernel size, we can solve the PSEUDO-ACHROMATIC NUMBER problem by enumerating all possible assignments of the vertices in the graph to the k groups, then checking whether any such assignment yields a valid grouping. We have the following corollary:

Corollary 3.10 *The* PSEUDO-ACHROMATIC NUMBER *problem can be solved in time $O(k^{(k-2)(k+1)} \cdot m\sqrt{n})$, and hence is fixed-parameter tractable.*

4 Hardness results for the VERTEX GROUPING problem

Recall from Section 1 that in the VERTEX GROUPING problem we are given an instance (G,H,k), where G and H are two graphs, and $k=|V(H)|$, and the problem asks for the existence of a surjective function $f: V(G) \longrightarrow V(H)$ satisfying the property that $\forall u,v \in V(H), (u,v) \in E(H)$ if and only if $\exists x \in f^{-1}(u), y \in f^{-1}(v)$ such that $(x,y) \in E(G)$. The VERTEX GROUPING problem can be defined more intuitively as follows.

Let G be an undirected graph. We define an operation on G, called *vertex grouping*, applied to a subset of vertices S as follows: remove all the vertices in S from G, add a new vertex w, and connect w to all the neighbors of S in $G-S$. The VERTEX GROUPING problem is:

> VERTEX GROUPING: Given two graphs G and H, where H is a graph of k vertices, and k is the parameter, decide if H can be obtained from G by a sequence of vertex grouping operations.

If H in the above definition is the complete graph on k vertices, then the VERTEX GROUPING problem becomes the PSEUDO-ACHROMATIC NUMBER problem, and hence is fixed parameter tractable. The following theorem shows that the VERTEX GROUPING problem is parameterized intractable in general.

Theorem 4.1 *(Theorem 5.1, Appendix) The* VERTEX GROUPING *problem is W[1]-hard.*

It was shown in [5] that, unless ETH fails, INDEPENDENT SET cannot be solved in time $n^{o(k)}$. Using the reduction from INDEPENDENT SET to VERTEX GROUPING given in the proof of Theorem 5.1 in the appendix, we can show that the same holds true for the VERTEX GROUPING problem:

Theorem 4.2 *Unless ETH fails, the* VERTEX GROUPING *problem cannot be solved in time $n^{o(k)}$, where n and k are the number of vertices in G and H, respectively.*

Determining the complexity of the GRAPH ISOMORPHISM problem is an outstanding open problem that has been attracting the attention of researchers in theoretical computer science for decades. Although no polynomial time algorithm was developed for the problem, it seems unlikely that the problem is NP-hard [11].

We illustrate a relationship between the GRAPH ISOMORPHISM problem and the VERTEX GROUPING problem. Let G_1 and G_2 be two graphs on n vertices. We are interested in knowing how "similar" G_1 and G_2 are, under the notion of vertex grouping defined above. For this purpose, we introduce the following parameterized problem:

> GRAPH STRUCTURAL SIMILARITY: given two graphs G_1 and G_2 on n vertices, and a parameter k, decide if there exists a graph H of k vertices such that both (G_1,H,k) and (G_2,H,k) are yes-instances of the VERTEX GROUPING problem.

Intuitively, the graph structural similarity measures the degree of similarity (i.e., k) between two graphs under the notion of vertex grouping. In particular, if $k = n$, then the GRAPH STRUCTURAL SIMILARITY problem is equivalent to the GRAPH ISOMORPHISM problem. We have the following parameterized intractability result for the GRAPH STRUCTURAL SIMILARITY problem:

Theorem 4.3 *(Theorem 5.2, Appendix) The* GRAPH STRUCTURAL SIMILARITY *problem is* $W[1]$*-hard.*

References

[1] R. Balasubramanian, V. Raman, and V. Yegnanarayanan, On the pseudoachromatic number of join of graphs, *International Journal of Computer Mathematics, 80-9*, 2003 , pp. 1131- 1137.

[2] V. Bhave, On the pseudoachromatic number of a graph, *Fundamenta Mathematicae, 102(3)*, 1979, pp. 159-164.

[3] H. Bodlaender, Achromatic number is NP-complete for cographs and interval graphs, *Information Processing Letters*, 32-3, 1989, pp.135-138.

[4] B. Bollobá, B. Reed, and A. Thomason, An extremal function for the achromatic number, *Graph Structure Theory*, 1991, pp.161-166.

[5] J. Chen, X. Huang, I. Kanj, and G. Xia, Strong computational lower bounds via parameterized complexity, *Journal of Computer and System Sciences, 72-8*, 2006, pp.1346-1367.

[6] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., McGraw-Hill Book Company, Boston, MA, 2001.

[7] R. Downey and M. Fellows, *Parameterized Complexity*, Springer-Verlag, 1999.

[8] K. Edwards and C. McDiarmid, The complexity of harmonious coloring for trees, *Discrete Applied Mathematics*, 57, 1995, pp.133-144.

[9] R. Downey, M. Fellows, and U. Stege, Parameterized Complexity: A Framework for Systematically Confronting Computational Intractability, in *Contemporary Trends in Discrete Mathematics*, (R. Graham, J. Kratochvíl, J. Nešetřil, and F. Roberts eds.), proceedings of the DIMACS-DIMATIA Workshop, Prague 1997, *AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 49*, (1999), pp. 49–99.

[10] R. P. Gupta, Bounds on the chromatic and achromatic numbers of complementary graphs. *In Recnet Progress in Combinatorics, Proceedings of Third Waterloo Conference on Combinatorics, Waterloo* (ed. W.T. Tutte), Academic Press, New York, 1969, pp. 229-235.

[11] J. Köbler, U. Schöning, and J. Torán, The Graph Isomorphism Problem: Its Structural Complexity, *Birkhäuser*, 1993.

[12] G. Kortsarz, J. Radhakrishnan, S. Sivasubramanian. Complete partitions of graphs, *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, 2005, pp. 860-869.

[13] E. Sampathkumar and V. Bhave, Partition Graphs and Coloring Numbers of Graphs, *Discrete Mathematics, 16*, 1976, pp.57-60.

[14] V. Yegnanarayanan, On pseudocoloring of graphs, *Utilitas Mathematica, 62*, 2002, pp. 199-216.

[15] V. Yegnanarayanan, The Pseudoachromatic Number of a Graph, *Southern Aian Bulletin of Mathematics, 24*, 2002, pp. 129-136.

5 Appendix

Theorem 5.1 *The* VERTEX GROUPING *problem is W[1]-hard.*

PROOF. We reduce the W[1]-hard problem INDEPENDENT SET to the VERTEX GROUPING problem.

Let (G, k) be an instance of the INDEPENDENT SET problem. Construct a graph G' by adding a new vertex w to G and connecting w to every vertex in G. Let H be a $(k+1)$-star with root r_H. Define the mapping π that, on an instance (G, k) of INDEPENDENT SET, produces the instance $(G', H, k+1)$ of VERTEX GROUPING. Clearly, the mapping π is computable in polynomial time, and hence π is an FPT-reduction. We show that (G, k) is an yes-instance of INDEPENDENT SET if and only if $(G', H, k+1)$ is an yes-instance of VERTEX GROUPING.

In effect, suppose that (G, k) is a yes-instance of INDEPENDENT SET, and let I be an independent set in G of size k. Consider the function $f : V(G') \longrightarrow V(H)$ that maps the k vertices of I in G' to the k leaves of the star H, in a one-to-one fashion, and maps all other vertices of G' to the root r_H of H. Then it is easy to verify that H is a vertex grouping of G' under the function f.

Conversely, suppose that H is a vertex grouping of G' under a function f. Consider any set of vertices I in G of cardinality k satisfying $f(I) = V(H) \setminus \{r_H\}$. Clearly, such a set I exists by the definition of the vertex grouping. Note that f is a bijection from I to $V(H) \setminus \{r_H\}$. Now for any two distinct vertices u and v of I, u and v are not adjacent in G, otherwise, by the definition of vertex grouping, $f(u)$ and $f(v)$ would be adjacent in H. It follows that I is an independent set of size k in G. This completes the proof. □

Theorem 5.2 *The* GRAPH STRUCTURAL SIMILARITY *problem is* $W[1]$*-hard.*

PROOF. As was shown in Theorem 4.1, the VERTEX GROUPING problem is $W[1]$-hard when the graph H is a star. An FPT-reduction can be constructed that takes an instance (G, H, k), where G has n vertices and H is a k-star, of the VERTEX GROUPING problem to an instance (G_1, G_2, k) of the GRAPH STRUCTURAL SIMILARITY problem, where $G_1 = G$ and $G_2 = H$. The $W[1]$-hardness of the GRAPH STRUCTURAL SIMILARITY problem follows. □