ON THE PROPER INTERVALIZATION OF COLORED CATERPILLAR TREES [*]

Carme Àlvarez[1] and Maria Serna[1]

Abstract. This paper studies the computational complexity of the *proper interval colored graph* problem (picg), when the input graph is a colored caterpillar, parameterized by hair length. In order prove our result we establish a close relationship between the picg and a graph layout problem the *proper colored layout problem* (pclp). We show a dichotomy: the picg and the pclp are NP-complete for colored caterpillars of hair length ≥ 2, while both problems are in P for colored caterpillars of hair length <2. For the hardness results we provide a reduction from the *multiprocessor scheduling problem*, while the polynomial time results follow from a characterization in terms of forbidden subgraphs.

Mathematics Subject Classification. 68Q25, 68W10.

1. Introduction

Interval graphs have been studied intensively because of their wide applicability to practical problems [11]. In the last years many efforts have been devoted to the study of problems in which one is asked to complete a graph into an interval graph. This kind of problems are used to model ambiguity in physical mapping or consistency in temporal reasoning [10]. One of them, the *interval graph completion* problem, asks whether it is possible to obtain an interval graph by adding at most k edges. This problem is known to be NP-complete (problem [GT35] of [8]). Another related problem is the *interval sandwich* problem: given two graphs $G_1 = (V, E_1)$

[1] ALBCOM Research Group, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord, Edifici Omega, C/ Jordi Girona Salgado 1-3, 08034 Barcelona, Spain; `alvarez@lsi.upc.edu`; `mjserna@lsi.upc.edu`

and $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$, decide whether there exist an interval graph $G = (V, E)$ where $E_1 \subseteq E \subseteq E_2$. This sandwich problem was introduced and shown to be NP-complete in [12]. A simpler proof of its NP-completeness is given in [9], where the authors extend the NP-completeness result to the case in which the sandwich graph is required to be a proper interval graph.

The colored versions of the *interval graph* problem and *interval sandwich* problem also have been studied. In such versions, the input is a graph together with a coloring of its vertices, and the solution is an interval graph or a sandwich graph, respectively, which is still properly colored by the input coloring. Indeed the *interval colored graph* and the *colored interval sandwich* problems are equivalent, in [6] and [9] it was proved their NP-completeness independently. In [2] it was shown that the *interval colored graph* problem is NP-complete even when it is restricted to the family of caterpillars. Two natural parameters have been considered in the colored version, the number of colors and the degree. For a fixed number of colors k and degree bounded by a constant, there is a $O(n^{k-1})$ algorithm to solve the *interval colored graph* problem [14]. For a fixed number of colors k, the problem is NP-complete for $k \geq 4$ and in P for $k < 4$ [3]. In fact for 4 colors the problem is NP-complete even for trees [1].

The problem that we will study here is another restriction of the above problems, the *proper interval colored graph* (PICG) problem: given a colored graph, can one add edges to obtain a proper interval graph that is still colored by the initial coloring. In [14] and [15] it was shown that the parameterized version of the PICG problem, with parameter the number of colors, is $W[1]$-hard, and this implies the NP-completeness of PICG. They also gave a polynomial time algorithm for constant number of colors. For another simpler NP-completeness proof for PICG see [7].

We concentrate here in the study of the computational complexity of the PICG, when the input graph is a colored tree. We show that the PICG is really hard as it is so even for caterpillars. We parameterize the caterpillar by hair length (see definitions later) and provide a complete computational complexity classification. The PICG is NP-complete for caterpillars with hair length ≥ 2, while for caterpillars with with hair length <2 it can be solved in polynomial time. Therefore the *proper interval colored graph* problem is as difficult as the *interval colored graph* on caterpillar trees. Notice that for, a bounded number of colors, the proper version can be solved in polynomial time, while the non-proper one is NP-complete for 4 colored trees.

To prove our results we make use of the close relationship between intervalizing problems and graph layout problems. The *interval graph completion* problem is equivalent to the *profile minimization* problem [13] also called the *MinSumCut* problem [4]. Moreover, the *interval colored graph* problem is equivalent to the *colored vertex separation* [2,5]. In this paper we define another graph layout problem, the *proper colored layout* problem (PCLP). Although the PCLP is not equivalent to the PICG, we show how to reduce the PICG problem to the PCLP problem, this will be enough for our purposes. The final result is the same dichotomy in the complexity: the PCLP is NP-complete for colored caterpillars of hair length ≥ 2

and in P for caterpillars of hair length 1 or 0. This contrast with the fact that the *interval graph completion* problem for trees is in P [13] and in NC [4].

The hardness results follow from a reduction from the multiprocessor scheduling problem, while the polynomial time algorithms are obtained from a characterization in terms of forbidden subgraphs.

2. DEFINITIONS AND BASIC RESULTS

A graph $G = (V, E)$ is called an *interval graph* if one can assign to each vertex $v \in V$ an interval on the real line I_v, in such a way that $(u, v) \in E \Longleftrightarrow I_u \cap I_v \neq \emptyset$. When in addition for any pair of vertices $u, v \in V$ it holds that $I_u \not\subseteq I_v$, the graph is called a *proper interval graph*. A *k-coloring* of a graph $G = (V, E)$ is a mapping $\kappa : V \longrightarrow \{1, \dots, k\}$. For any vertex subset $V' \subseteq V$, let $\kappa(V') = \{\kappa(v) \mid v \in V'\}$. A *proper coloring* of G is a coloring such that no two adjacent vertices have the same color. A *k-colored* graph is a graph together with a proper k-coloring. We consider the following intervalization problems for colored graphs:

> Interval colored graph (ICG)
> Given a k-colored graph $(G = (V, E), \kappa)$, decide whether there is an edge superset E' such that the graph $G' = (V, E')$ is an interval graph and κ is still a proper coloring of G'.

> Proper interval colored graph (PICG)
> Given a k-colored graph $(G = (V, E), \kappa)$, decide whether there is an edge superset E' such that the graph $G' = (V, E')$ is a proper interval graph and κ is still a proper coloring of G'.

The next technical result will be used later and follows from the definitions.

Lemma 2.1. *If a k-colored graph $(G = (V, E), \kappa)$ has a proper colored supergraph that is a proper interval graph, then any of its subgraphs verifies the same property.*

Let us now introduce some colored versions of graph layout problems. Given a graph $G = (V, E)$ with $|V| = n$ a *layout* (or *linear ordering*) φ of G is a one to one mapping $\varphi : V \longrightarrow \{1, \dots, n\}$. For any layout φ of G, and any i, with $1 \leqslant i < n$, we define the set $V_i = \{v \mid \varphi(v) \leqslant i \text{ and } \exists u \ \varphi(u) > i \ (u, v) \in E\}$. For a given k-colored graph $(G = (V, E), \kappa)$, a *colored layout* of (G, κ) is a layout φ of G such that for all $u \in V$ with $\varphi(u) > 1$, $\kappa(u) \notin \kappa(V_{\varphi(u)-1})$. This is equivalent to say that for any $u, x, v \in V$, if $(u, v) \in E$ and $\varphi(u) < \varphi(x) < \varphi(v)$ then $\kappa(u) \neq \kappa(x)$. For a given k-colored graph $(G = (V, E), \kappa)$, a *proper colored layout* of (G, κ) is a colored layout φ of G such that for all $u, v \in V$ and $x \in V$ with degree at least 2, if $(u, v) \in E$ and $\varphi(u) < \varphi(x) < \varphi(v)$ then it exists a vertex y such that $\varphi(v) < \varphi(y)$, $(x, y) \in E$. We consider the following layout problems for colored graphs:

> Colored layout problem (CLP)
> Given a k-colored graph (G, κ), decide whether there is a colored layout φ of (G, κ).
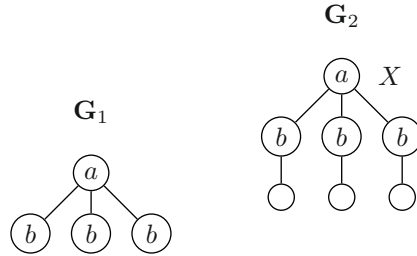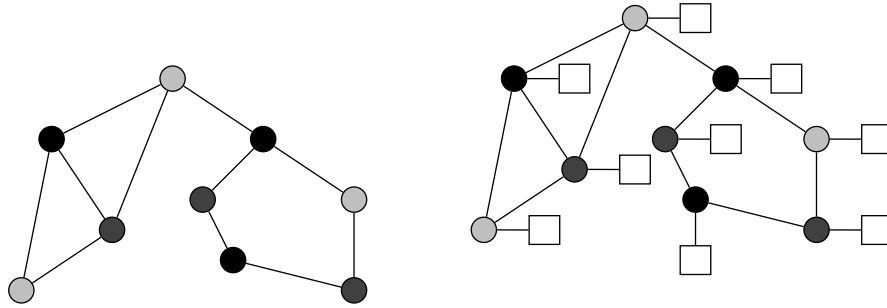
FIGURE 1. Some graphs.

FIGURE 2. A graph G and a decoration of G.

Proper colored layout problem (PCLP)
Given a k-colored graph (G, κ), decide whether there is a proper
colored layout φ of (G, κ).

It is known that the CLP problem is identical to the ICG problem [2,5]. For the
proper case this result does not hold. For example, the graph G_1 given in Figure 1
has a proper colored layout, but does not have any proper intervalization (a label
inside a circle indicates a color, while a label outside a circle, if any, indicates a
node name). However, we will show that the PICG problem can be formulated as
an instance of the PCLP, for a particular class of graphs.

Given a k-colored graph $(G = (V, E), \kappa)$, a decoration of G is a new $(k + 1)$-
colored graph $(G^+ = (V', E'), \kappa')$ (see Fig. 2), in which we attach exactly one new
vertex to each vertex. All the new vertices have degree one and are colored with
the same color, a new color which is not used by κ. Formally, $V' = V \cup V^+$ where
$V^+ = \{u^+ \mid u \in V\}$, that is, a marked copy of V. The set of edges is $E' = E \cup E^+$
where $E^+ = \{(u, u^+) \mid u \in V\}$, and for any u in V, $\kappa'(u) = \kappa(u)$ and $\kappa'(u^+) = c$,
where c is a new color, therefore κ' is a $k + 1$ coloring of G^+.

Theorem 2.2. $(G, \kappa) \in$ PICG *if and only if* $(G^+, \kappa') \in$ PCLP.

Proof. First, notice that a non connected graph has a proper colored layout if and only if each of its connected components has one, furthermore, a proper colored layout (or a proper intervalization) of the graph can be obtained solving the corresponding problem on each of its components. Second, in the case that G is an isolated vertex, the graph is a proper interval, and the corresponding decorated graph also has a proper colored layout.

Now, we will prove the result for a connected graph G with more than one vertex. Assume that $\{I_u\}_{u \in V}$ is a proper intervalization of $G = (V, E)$, where each interval is of the form $I_u = [a_u, b_u]$. We assign to each node u two integers,

$$n_1(u) = \#\{a_v \mid a_v \leq a_u\} + \#\{b_v \mid b_v < a_u\} \text{ and}$$
$$n_2(u) = \#\{a_v \mid a_v < b_u\} + \#\{b_v \mid b_v \leq b_u\}.$$

Now we define a layout for G^+, $\varphi(u) = n_1(u)$ and $\varphi(u^+) = n_2(u)$. Since we start from a proper intervalization it follows that φ is a proper colored layout of G^+.

Assume that φ is a proper colored layout of G^+. For each node $u \in V$ we define $n(u) = \max\{\varphi(v) \mid (u, v) \in E'\}$, notice that φ is a proper colored layout of G^+, therefore we always have $\varphi(u) < n(u)$ for any node $u \in V$. Assigning the interval $[\varphi(u), n(u)]$ to u we get a proper intervalization of G. $\square$

For sake of simplicity we will use the term *decorated graph* to denote a $k+1$-colored graph in which there is a color c, for which, all the nodes with degree one are colored with color c, and any vertex that has a color different from c has exactly one neighbor colored c. In other words a decorated graph is a decoration of the graph obtained by removing all vertices colored with c.

3. Some caterpillars and decorated caterpillars

A caterpillar with hairs of length at most h is formed by a chain, called the *backbone*, each node in the backbone can be connected to several non-intersecting paths of length at most $h - 1$. A k-*star* with hair length h is a single node, called the *center*, connected to k paths of length $h - 1$, see Figure 1.

In this section we study the existence of proper colored layouts for some particular cases of caterpillars and decorated caterpillars. Notice that a decorated caterpillar of hair length 2 or more is not a caterpillar. The graphs introduced here will be used in the following sections as part of other graph gadgets.

Lemma 3.1. *A k-star with hair length h, such that all the neighbors of the center have the same color, has no proper colored layout for any $k \geq 3$ and $h \geq 2$, independently of the colors assigned to the remaining nodes.*

Proof. According with Lemma 2.1 it is enough to prove it for the case $k = 3$ and $h = 2$. Assume that the center X is colored with color a and its three children with color b (see the graph G_2 given in Fig. 1).
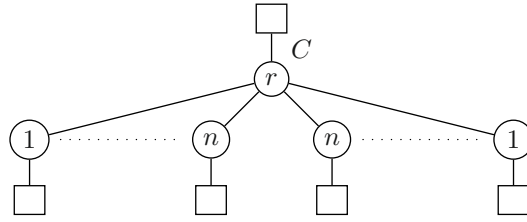
FIGURE 3. A barrier $B(\Delta, r)$, with $\Delta = \{1, \ldots, n\}$.

Two of the b-colored nodes cannot lie to the left of X, one will block the other color. On the other hand, two of the the b-colored nodes cannot lie to the right of X, notice that after placing one, before the other, we need to put the corresponding leaf, thus obtaining a colored layout that is not proper. □

Our next graph is the colored decorated caterpillar given in Figure 3. The graph is a decoration of a $2k$-star with hair length 1, whose center will be in the backbone of the caterpillar. The graph is colored with $k+1$ colors, in such a way that the center has exactly two children with the same color. We call such graph a *barrier* with children's colors Δ and *center* colored r for some color $r \notin \Delta$. We will denote such graph by $B(\Delta, r)$. All decorations (empty squares in the figure) will be colored with the same color, with the only required condition that the coloring must be proper. The main properties of a barrier are the following.

Lemma 3.2. *In any proper colored layout φ of a barrier $B(\Delta, r)$ the children must be partitioned into two sets, one in each side of the center in such a way that in each set there is no repeated color.*

Proof. Let us consider two children of the center, with identical color $i \in \Delta$. Both children can not lay in the left side of the center because one will block the color of the other one. Neither they can lay on the right side of the center because, once we have placed the first son, its decoration must be placed before the second child, otherwise the color i will be blocked. Therefore any colored layout will not be proper. □

The previous lemma implies that in any proper colored layout of a barrier, the children of the center must be splited into two sets, one appears before the center and the other after the center. Notice that none of the two sets has two vertices with the same color. In such a situation the corresponding decorations must be placed to the right, as it is shown in Figure 4.

Notice that our barrier is fully decorated, for sake of simplicity we shall omit some of the decorations, when in the proposed layout such decorations can be placed just right after the node.

Lemma 3.3. *Let $B = B(\Delta, r)$ a barrier and let $e = (u, v)$ be a decorated edge such that u and v do not form part of the barrier and $\kappa(\{u, v\}) \subseteq \Delta \cup \{r\}$. Then,*
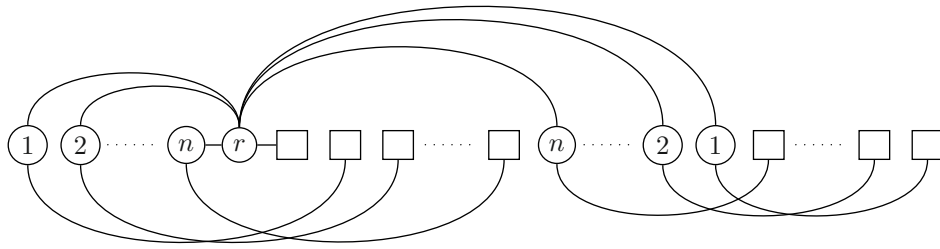
FIGURE 4. A proper colored layout for the barrier $B(\Delta, r)$.

in any proper colored layout of the barrier and the edge, the two endpoints of the edge must lay both either before or after any of the barrier's nodes.

Proof. First let us consider the case that $\kappa(u)$ and $\kappa(v)$ are different from r the color of the center c. Then there are four nodes x, x', y, y' of the barrier such that $\kappa(x) = \kappa(x') = \kappa(u)$ and $\kappa(y) = \kappa(y') = \kappa(v)$. In this case it is easy to see that u and v must be placed in the same side with respect to c. Otherwise, if $\varphi(u) < \varphi(c) < \varphi(v)$ then u will block or will be blocked by x or x'. And if we interchange u by v then v will block or will be blocked by y or y'.

And second let us consider that u has the same color than the center c. If $\varphi(u) < \varphi(c)$ then $\varphi(v) < \varphi(c)$ otherwise u will block the color of the center. And if $\varphi(c) < \varphi(u)$ then u must be placed after the children of c once the color of c is free. Moreover v must be placed once its color is free, that is after placing the decoration of the son of the barrier with the same color.

Up to now we have that both u and v have to lay in the same side of c. In the case $\varphi(u) < \varphi(v) < \varphi(c)$, the decoration of v must be placed to the right of it in order to get a proper layout. Then v must be placed before any children of c, otherwise the color of v will block or will be blocked by the node of the barrier with identical color. And for the case $\varphi(c) < \varphi(u) < \varphi(v)$ we have that u must be placed in the rightmost part with respect to the children of c because it will block or it will be blocked by the node of the barrier with the same color (either c or x). $\qquad\square$

The next lemma gives the conditions that an edge need to verify, in order to jump a barrier.

Lemma 3.4. *Let $B = B(\Delta, r)$ be a barrier and let $e = (u, v)$ be a decorated edge edge such that u and v do not form part of the barrier. Then, there is a joint layout of the barrier and the edge in which the barrier's center is between u and v if and only if $\kappa(\{u, v\}) \cap (\Delta \cup \{r\}) = \emptyset$.*

Proof. By the previous lemmas we know that if $\kappa(\{u, v\}) \subseteq \Delta \cup \{r\}$ then u and v must be placed just before or after any of the children of the center. It only remains to show that when only one of u and v has color in the set $\Delta \cup \{r\}$, both nodes must lay in the same side of the center. Assume that $\kappa(v) \in \Delta \cup \{r\}$ then v
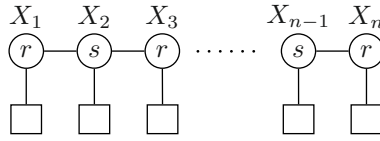
FIGURE 5. Alternating chain $C_n(r, s)$.

must be placed just before or after the children of c. If v is placed to the left of c then u must also be placed to the left of c, otherwise v will block the color of one node in the barrier. If v is placed to the right of c then u must also be placed to the right of c, otherwise the layout will not be proper. □

The second caterpillar that we consider is the decorated alternating chain given in Figure 5. This chain has n nodes in its backbone, odd position nodes are colored with the same color r, and even position nodes are colored with another color s, $r \neq s$. We call such a chain an *alternating chain*, denoted by $C_n(r, s)$. The main properties of an alternating chain are described in the following lemmas. The first is that no proper colored layout can turn the relative positions in the chain.

Lemma 3.5. *In any proper colored layout of an alternating chain $C_n(r, s)$, the backbone nodes must be laid out as they are.*

Proof. Just notice that any subgraph formed by three consecutive backbone nodes, and their corresponding decorations is a barrier. □

The second property concerns the joint layout of an alternating chain and an edge, with different colors.

Lemma 3.6. *In any proper colored layout of an alternating chain $C = C_n(r, s)$ and a decorated edge $e = (u, v)$ such that $\kappa(\{u, v\}) \cap \{r, s\} = \emptyset$, no two backbone nodes of C_n with the same color can be placed between u and v.*

Proof. The previous lemma implies that the chain must be laid in order, let us assume that it corresponds to the order given in Figure 5. If we place the nodes u and v in such a way that the edge covers at least two consecutive chain nodes with the same color, X_i and X_{i+2} then we have $\varphi(u) < \varphi(X_i) < \varphi(X_{i+1}) < \varphi(X_{i+2}) < \varphi(v)$ for some i. In this case the decoration of X_i must be placed before X_{i+2} otherwise the color of X_{i+2} will be blocked by X_i. But, such colored layout is not proper. □

As a consequence of the previous lemmas we can state under which conditions some alternating chains can be laid together.

Since any two identical colored nodes of the backbone of an alternating chain cannot be placed between the two ends of any decorated edge, we have that at most two adjacent chain nodes can be placed between the two ends of a decorated edge.

Corollary 3.7. *Let $C_1 = C_n(r, s)$ and $C_2 = C_m(p, q)$ be two alternating chains, such that $\{r, s\} \cap \{p, q\} = \emptyset$. Then, there is a joint proper colored layout of C_n*
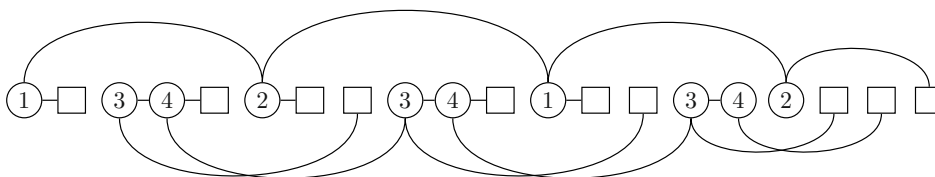
FIGURE 6. A joint layout of $C_4(1,2)$ and $C_6(3,4)$.

and C_m in which the first and the last node are the end points of C_n, if and only if $m \leq 2(n-1)$.

Moreover, if we consider two identical decorated edges, the only possible joint layout must place one edge after the other, with no overlap.

Corollary 3.8. *Let $C_1 = C_n(r,s)$, $C_2 = C_m(p,q)$, and $C_3 = C_l(p,q)$ be three alternating chains such that $\{r,s\} \cap \{p,q\} = \emptyset$. Then, there is a joint proper colored layout of C_n, C_m and C_l in which the first and the last node are the end points of C_n, if and only if $m + l \leq 2(n-1)$.*

Proof. Notice that any joint proper colored layout of two alternating chains with identical colors must place one entire chain completely before the other starts. This happens because no node of the backbone of one chain can be placed between two nodes of the other since at least two of them have the same color. Hence, by using this fact and the previous lemma the result holds. $\square$

We will also consider another caterpillar, that we will call a *generalized alternating chain*, as given in Figure 7. This chain has $2n+1$ nodes in its backbone, odd position nodes are colored with the same color r, and even position nodes may have different color and can be colored with any color different from r. Again, no proper colored layout of a generalized alternating chain can turn the relative positions in the chain.

Lemma 3.9. *In any proper colored layout of a generalized alternating chain, the backbone nodes must be laid out as they are.*

Proof. Assume that we have a generalized alternating chain as the one given in Figure 7. First notice that any subgraph formed by a backbone node in an even position and its two adjacent backbone nodes, with their corresponding decorations, is a barrier. Hence, it is not possible to turn out the backbone in an even position node.

The other possibility is to turn out the backbone in an odd position node Y which has as adjacent nodes X_i and X_{i+1}. Notice that X_i and X_{i+1} are centers of barriers in which the node Y is one common child of them. Hence if we consider that $\varphi(Y) < \varphi(X_i) < \varphi(X_{i+1})$, the two backbone neighbors of X_i and X_{i+1} with the same color of Y must be placed to the right of X_{i+1}. But such a layout can not be proper. And if we consider the symmetric case, $\varphi(X_i) < \varphi(X_{i+1}) < \varphi(Y)$
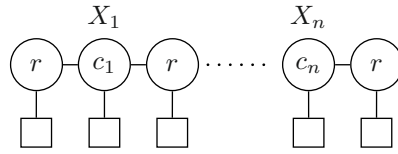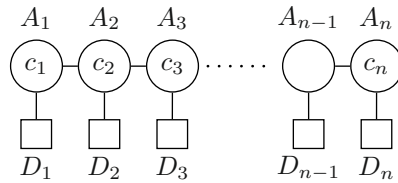
FIGURE 7. A generalized alternating chain.

FIGURE 8. A zig-zag chain.

we have that the two backbone neighbors of X_i and X_{i+1} must be placed to the left and then one will block the other's color. □

Now we analyze another chain, in which instead of having alternating colors on the backbone, each node in the backbone has a completely different color. This chain has n nodes in its backbone with colors $\Delta = \{c_1, \ldots, c_n\}$. We call such a chain a *zig-zag chain*, denoted by $Z_n(\Delta)$, see Figure 8. One of the main properties of a zig-zag chain is that it can be turned at any position.

Lemma 3.10. *Let π be a permutation of the set $\{1, \ldots, n\}$, for some $n \geq 2$, and let $Z = Z_n(\Delta)$ be a zig-zag chain. Then, the layout φ defined as $\varphi(A_i) = \pi(i)$ and $\varphi(D_i) = n + \pi(i)$ for $1 \leqslant i \leqslant n$ is a proper colored layout of $Z_n(\Delta)$.*

Proof. As all the colors are different, and backbone nodes are placed before any decoration, the layout φ is a colored layout. To see that it is also proper, just take into account that the relative order of the backbone nodes in φ is preserved in its decorations. □

Next we study the shrink properties of a zig-zag chain in a joint layout with an additional edge. Extending the argument of the previous lemma we get,

Lemma 3.11. *Let $Z_n(\Delta)$ be a zig-zag chain, and $e = (u, v)$ a decorated edge, such that $\kappa(\{u, v\}) \cap \Delta = \emptyset$. For any $n \geq 2$, and any given permutation π on the set $\{1, \ldots, n\}$, the layout φ defined as $\varphi(u) = 1$, $\varphi(v) = n + 2$, $\varphi(A_i) = \pi(i) + 1$ and $\varphi(D_i) = n + \pi(i) + 2$ for $1 \leqslant i \leqslant n$ is a joint proper colored layout of $Z_n(\Delta)$ and e.*

The next lemma gives the stretch properties of a zig-zag chain in a joint layout with an alternating chain. Its proof follows from Lemmas 3.5 and 3.6.

Lemma 3.12. *Let $C_n(r, s)$ be an alternating chain and $Z_m(\Delta)$ a zig-zag chain, with $\{r, s\} \cap \Delta = \emptyset$. There is a proper colored layout of both graphs, in which the*
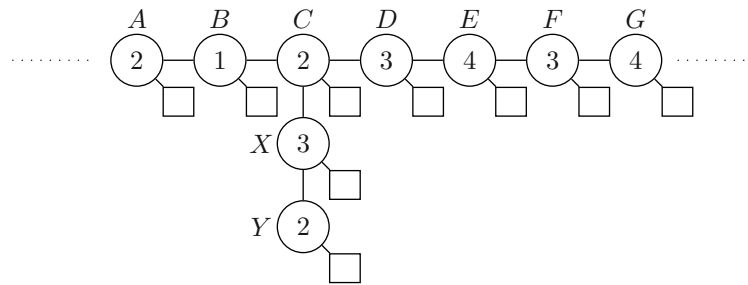
FIGURE 9. The turning point T.

first and the last nodes that are not decorations, are from the backbone of Z_m if and only if $n \leqslant 2(m-1)$.

4. THE REDUCTION FOR THE PICG PROBLEM

We will show that the PICG problem is NP-complete for caterpillars with hairs of length two. Our proof has been inspired in the schema used in [16] to show that the bandwidth problem is NP-complete for caterpillars. We give a reduction from the *multiprocessor scheduling problem*. That is, given a set of n tasks, having duration t_i for $1 \leqslant i \leqslant n$, a deadline D and a set of m processors, determine whether the tasks can be assigned to processors so that all processors finish their work before the deadline D. Recall that the problem is NP-complete in the strong sense, so we can assume that the duration of each task is polynomially bounded.

Our reduction will produce a decorated tree (in fact a decorated caterpillar) such that it has a colored layout if and only if the original multiprocessor scheduling instance has solution. Our graph is a decoration of a colored caterpillar of hair length 2, therefore we will show that the PCLP problem is NP-complete for decorations of caterpillars of hair length 2, and therefore we show that the PICG problem is NP-complete for caterpillars of hair length 2.

The first gadget is the colored caterpillar T given in Figure 9, the dashed lines represent the connection to other gadgets to be described later. We will refer to such caterpillar as *the turning point* with center C. Our construction of the caterpillar is based in the fact that in any colored layout of the turning point, the two end nodes of the backbone (A and G) must be laid in the same side of the center C.

Theorem 4.1. *In any proper colored layout of the turning point T, the nodes A and G must be laid both on the same side of C.*

Proof. The subgraphs induced by $\{A, B, C, X, Y\}$ and $\{X, C, D, E, F\}$ are generalized alternating chains, so any proper layout must preserve its relative order. Nodes X and D cannot go to the same side of C, because they have the same color. Therefore the only possibility is the one stated in the theorem.   □
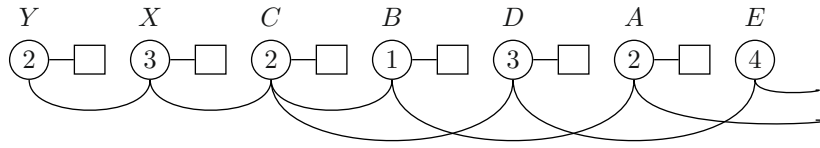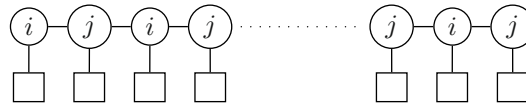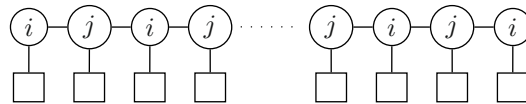
FIGURE 10. A proper colored layout for the turning point.

FIGURE 11. The graph P associated to one processor.

According to the previous theorem and the results on chains, the layouts for the turning point T, must follow the lines of the one given in Figure 10. The two chains on the right may interchange order, however, the connections to other gadgets will force the showed order. It is also possible a symmetric layout in which the order of the backbone nodes is reversed and some of the decorations are moved to the right of their corresponding node.

A processor is represented by an alternating chain $C_{D+1}(1,2)$ or $C_{D+1}(2,1)$, depending on the processor number and the parity of D, as given in Figure 11. The graph R_m, associated to the m processors is formed by series of barriers, separating the graphs corresponding to processors i and $i+1$. There are also two additional big barriers, one at the beginning and the other at the end. (See the left part of the caterpillar given in Fig. 13.) The number of edges in the backbone between two successive barriers is D. The little barriers connecting the alternating chains are $B(\{2,3,4\},1)$ or $B(\{1,3,4\},2)$ depending on the parity of D and on the position of the ith processor. Notice that the children with color 2 or 1 (respectively) form part of the backbone. The first big barrier is connected with the end node A of the turning point. It is a $B(\Delta \cup \{2\},1)$ barrier where Δ is
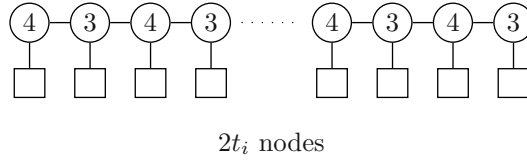
2t_i nodes

FIGURE 12. The alternating chain associated to task i.

the set of colors that we will use for connecting the gadgets representing the tasks (Δ does not contain colors 1, 2, 3 and 4), and again the children colored 2 appear in the backbone. The last barrier is either $B(\Delta \cup \{2,3,4\}, 1)$ or $B(\Delta \cup \{1,3,4\}, 2)$ depending on the parity of D and of m. Notice that this barrier contains all colors used in the final decorated caterpillar produced by the reduction.

Lemma 4.2. *In any proper colored layout of the graph R_m the backbone must be laid out as it is, between the centers of the first and the last barriers.*

Proof. The backbone of R_m is an alternating chain and some of its nodes are centers of barriers. Then by Lemmas 3.2 and 3.5 the backbone must be laid as it is, furthermore, the children of each barrier must be partitioned into two sets one in each side of the backbone node corresponding to their corresponding center. $\square$

Each task i is represented by an alternating chain $T_i = C_{2t_i}(3,4)$. (See the right part of the caterpillar given in Fig. 13.) We will join the graphs corresponding to tasks i and $i+1$ ($1 \leqslant i < n$) with a zig-zag chain $Z_l(\Delta_{i+1})$ of length equal to the length of the backbone R_m ($l = m(D+1) + m - 1$). We select a set of new different colors Δ_{i+1} for each connection between tasks i and $i+1$. An additional zig-zag chain $Z_l(\Delta_1)$ joins the first task with the node G of the turning point. For $1 \leqslant i, j \leqslant n$, $\Delta_i \cap \{1,2,3,4\} = \emptyset$ and if $i \neq j$ then $\Delta_i \cap \Delta_j = \emptyset$. Let $\Delta = \cup_{i=1}^n \Delta_i$.

Given an instance $I = (t_1, \ldots, t_n, D, m)$ of the multiprocessor scheduling problem let $G(I)$ be the colored decorated caterpillar obtained by joining the three gadgets through the corresponding vertices (see Fig. 13).

Theorem 4.3. *The instance $I = (t_1, \ldots, t_n, D, m)$ of multiprocessor scheduling has a solution if and only if the colored decorated caterpillar $G(I)$ has a proper colored layout.*

Proof. From the previous lemmas the only possible colored layout for the caterpillar must maintain as it is the ground line corresponding to the processors, between the center of the two big barriers, the turning point must be *turned* at its center C and the two endpoints A and G must be placed in the same side with respect to C (by Thm. 4.1). Notice that A is connected to the processors gadget, with a barrier $B(\Delta \cup \{2\}, 1)$ which does not contain $\{3,4\}$. On the other hand G is connected to the tasks gadget, with a zig-zag chain of colors $\Delta_1 \subseteq \Delta$. Therefore, by Lemma 3.4, the edges (E, F) and (F, G) can jump the barrier while the edge formed by G and the first node of the zig-zag chain cannot cross the barrier. Hence G must be placed just after (or just before depending if we consider the turn to

Processors R_m Tasks

① ☐

② $B(\Delta \cup \{1,3,4\},2)$

① ☐   ④ ☐

P_m $Z_l(\Delta_n)$

① ☐

② $B(\{3,4\},2)$
① ☐ $Z_l(\Delta_2)$

① ☐
② $B(\{3,4\},2)$
① ☐   ③ ☐

P_1 T_1

① ☐   ④ ☐

②

A' ① $B(\Delta,1)$ $Z_l(\Delta_1)$

A ② — ① — ② — ③ — ④ — ③ — ④ G
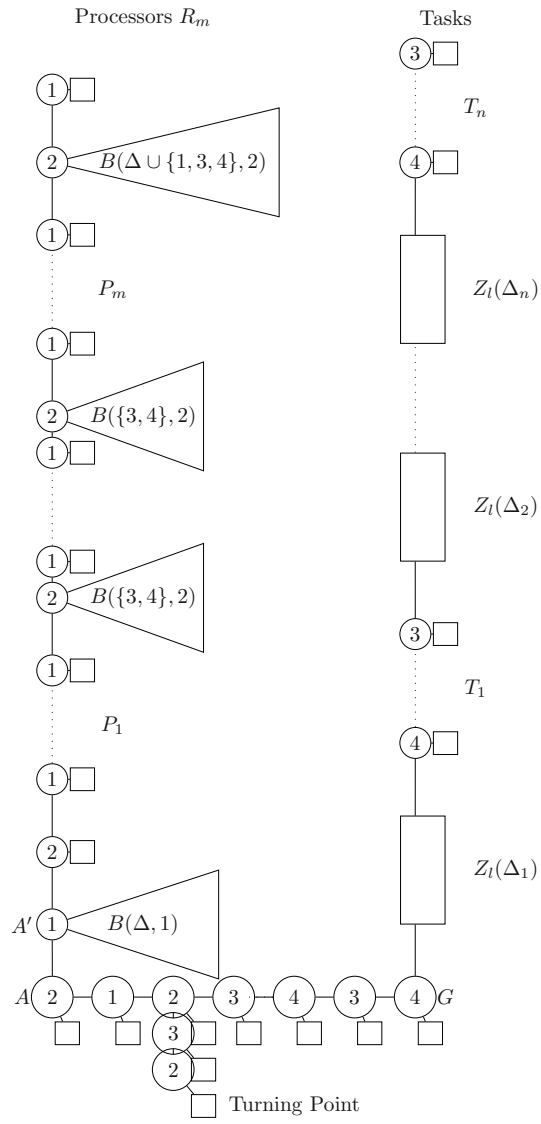☐   ☐   ③   ☐   ☐   ☐   ☐
        ②
        ☐ Turning Point

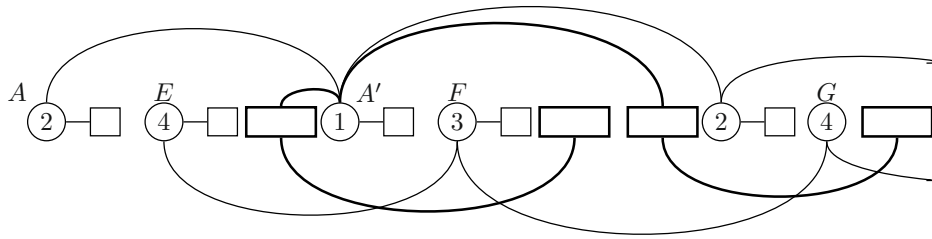FIGURE 13. A sketch of the caterpillar $G(I)$.

FIGURE 14. A detail of the layout at the end of the turning point.

the right or to the left) the child of the barrier attached to the backbone which connects the barrier with the processors gadget. (See Fig. 14.) In such a situation the first zig-zag chain of the tasks gadget must be placed just after (or just before, depending on the direction of the turning) the first barrier in the processors.

The piece corresponding to the tasks must be placed between the center of the two barriers of the processors part. Now from Lemma 3.3 the alternating chain corresponding to just one task, must go between the center of two barriers. And from Lemma 3.8 the total duration of the tasks placed in the line corresponding to a processor must be at most D. Finally notice that the shrink and stretch properties of the zig-zag chains given in Lemmas 3.11 and 3.12, allow us to place different tasks in any relative position. So each scheduling meeting the deadline requirement gives a proper colored layout and every proper colored layout gives a correct scheduling.   □

Notice that in our reduction we construct a decorated caterpillar with $p(n)$ vertices, and $q(n)$ colors, where p and q are polynomials, and n is the number of tasks in the original instance. Our original caterpillar has all hairs with length 1 except for one that have length 2. That makes our gadget not a caterpillar.

Theorem 4.4. PCLP *is NP-complete for decorated caterpillars with hair length 2. Therefore* PICG *is NP-complete for caterpillars with hair length 2.*

The previous reduction can be modified to show that the PCLP problem is NP-complete for caterpillars with hair length at most 2. The changes affect the turning point and the task's gadget. The new turning point, the graph T' given in Figure 15, is obtained by removing the decorations of nodes X and Y, adding an additional hair identical to the hair (X, Y), and interchanching some colors. The new task gadget is obtained from the one given in Figure 12 interchanging the colors 3 and 4.
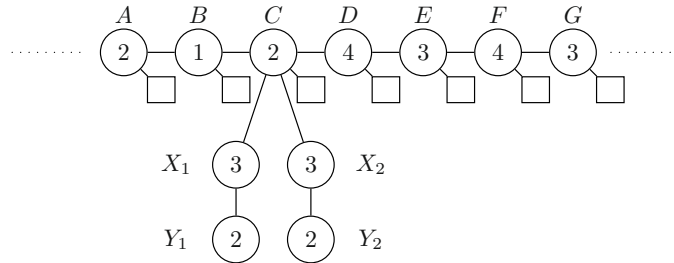
FIGURE 15. The turning point T' for the reduction from multi-processor scheduling to the PCLP.

Theorem 4.5. *In any proper colored layout of the turning point T', the nodes A and G must be laid both to the same side of C.*

Proof. First notice that the subgraph induced by nodes $\{C, X_1, Y_1, X_2, Y_2\}$ is a barrier with center C, therefore, X_1 and Y_1 cannot be placed in the same side X_2 and Y_2 with respect to C.

Let us suppose that $\varphi(Y_2) < \varphi(X_2) < \varphi(C) < \varphi(X_1) < \varphi(Y_1)$. The decorated edge (E, D) shares one color with the barrier, therefore by Lemma 3.4, both nodes must be placed in the same side with respect to C. As D is connected to C and, nodes X_1 and Y_1 are not decorated, the only possibility is to place D between X_1 and Y_1 and E after Y_1. The same argument, applied to the decorated edge (B, A), forces the placement of B between X_1 and Y_1 and of A after Y_2. $\square$

Using as turning point the graph T' together with the other gadgets, and connecting them in the same way, we obtain a colored caterpillar with hair length two. Notice that from the previous theorem there are only two possible layouts for the turning point, depending on whether $\varphi(C) < \varphi(B) < \varphi(D) < \varphi(A) < \varphi(E)$ or $\varphi(C) < \varphi(D) < \varphi(B) < \varphi(E) < \varphi(A)$. The second possibility will not allow a proper colored layout, when we join the node A with the first barrier in the processors gadget, for this part we are again in the situation of Figure 14. Therefore,

Theorem 4.6. PCLP *is NP-complete for caterpillars with hair length 2.*

5. SHORT HAIRED CATERPILLARS

In this section we will show that for decorations of caterpillars with hair length 1 or 0, and for caterpillars with hair length 1 or 0, the PCLP is in P, in fact in NC. A caterpillar with hairs of length 0 is just a line, therefore for any given proper coloring, the line structure gives a proper colored layout. A caterpillar with hairs of length 1 is just a linear sequence of stars, provided that the coloring is proper,

we can lay out the backbone, following the linear order. Then place all the degree one nodes just right after the star center. When we decorate a caterpillar with hairs of length 0 we get another caterpillar with hairs of length 1, therefore:

Lemma 5.1. *The* PCLP *is in NC, for caterpillars of hairs of length 0 and 1. The* PICG *is in NC, for caterpillars of hairs of length 0.*

Let us now give an algorithm for the case of decorations of caterpillars with hair length 1. First we consider the decorated caterpillars given in Figure 16. The central caterpillar has no proper colored layout, while the other two have. Furthermore, the layout will place all the backbone nodes in order, so that each backbone node is preceded and followed by nodes colored with color b.

Lemma 5.2. *For any $k \geq 1$, any proper colored layout of the decorated caterpillar D_k, given in Figure 16, must lay all the backbone vertices in the given order (either increasing or decreasing), a node colored b before any of the backbone nodes, a node colored b after all the backbone nodes and a node colored b between any pair of consecutive backbone nodes.*

Proof. First notice that the graph has two proper colored layouts, verifying the required properties, one for each of the two possible orderings of the backbone nodes. Following the order of backbone nodes, interleaving the b nodes and placing the decorations to the right.

Our proof that every layout is of the proposed form is by induction on k. For $k = 1, 2$ any proper colored layout verify the properties.

Assume that for $k = n$ the condition hold. Assume that φ is a proper colored layout of D_{n+1}. Notice that D_{n+1} can be seen as a graph D_n with an additional path. We consider two cases:

Case 1. A_{n+1}, in φ, does no appear as the last neighbor of A_n.
Let x, y, z be three nodes in the subgraph D_n such that $(x, z) \in E$ and $\varphi(x) < \varphi(y) < \varphi(z)$, as φ is a proper colored layout there must be a node w connected to y laid out after z. When $y \neq A_n$ such z node must belong to D_n. But when $y = A_n$ and $w = A_{n+1}$, by the condition there is another neighbor of y laid after w. Therefore there is always a node $z \in D_n$ connected to y and laid out after z. Therefore the layout φ' obtained eliminating the node A_{n+1}, its hair and its decoration, and compacting numbers when necessary, is a proper colored layout of C_n.

Case 2. A_{n+1}, in φ, appears as the last neighbor of A_n.
There are two possibilities, there is a decoration placed between A_n and A_{n+1} connected to a node on the right of A_{n+1} or not. In the first subcase move the decoration of node A_n to the position just before this decoration. In the second move the decoration of A_n to A_{n+1}´s place. After that, by removing A_{n+1}, its hair and its decoration, compacting numbers when necessary, we obtain a proper colored layout φ'' of D_n.

In both cases the relative order of the nodes is the same as in φ, therefore, by induction the layout φ must place in order (ascending or descending) the nodes
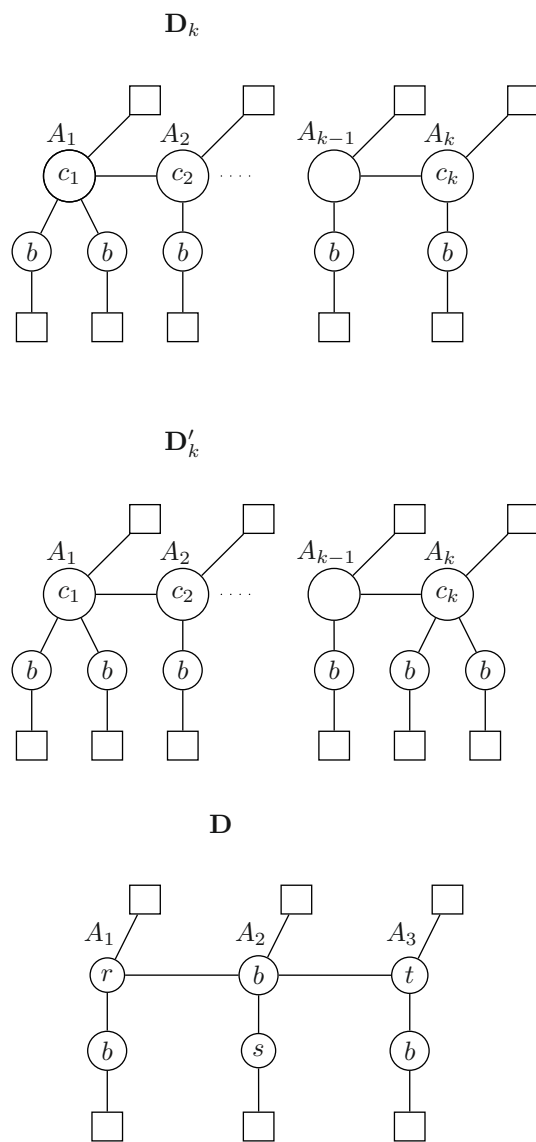
D$_k$

D$'_k$

D

FIGURE 16. Some decorated caterpillars.

A_1 to A_n and interleave the b nodes. But in such situation the only possibility for A_{n+1}, connected to A_n is to be placed following the sequence. Therefore φ verifies the hypothesis. $\qquad\square$

By adding an additional b-colored hair, we obtain a caterpillar with no proper colored layout.

Lemma 5.3. *For any $k \geq 2$, the graph D'_k, given in Figure 16 has no proper colored layout.*

Proof. By removing one of the b-colored neighbors of A_n we get a D_k graph. Using the previous lemma, in any proper colored layout of D_k there is a b-colored node before A_n and another after. Therefore it is not possible to place an additional b-colored node connected to A_n. $\qquad\square$

Notice that the graph D, given in Figure 16, when the colors r, s, t are not all the same, has a proper colored layout. Therefore this configuration can be the start and/or end of a D_k or D'_k subgraph.

Theorem 5.4. *Any decorated caterpillar with hair length 1 that contains neither any instance of the subgraphs D'_k, given in Figure 16, nor the subgraph G_2, given in Figure 1, has a proper colored layout.*

Proof. We will give a procedure to obtain a proper colored layout in the absence of the cited subgraphs. Our algorithm will place the backbone in order. Assume that we have fixed one of the two possible orderings, denoted by π.

Now fix any color b. We first identify all nodes in the backbone with two children colored b. From that nodes we isolate all the subgraphs of the form of D_k exploring to the right and to the left. All nodes colored b in a D_k subgraph can be classified as *left* or *right*, by Lemma 5.3, depending on whether the backbone node with two b-colored children is on its right or on its left according to π. For the two sons with the same color b one will be declared *left* and the other one *right*. The remaining nodes, not in the backbone, that have not been classified will be classified as *right*.

Now, for every node u in the backbone, every non backbone node connected to u has been classified as left or right. Let us denote by $L(u)$ the set of left neighbors, and by $R(u)$ the set of right neighbors.

From the previous classification, and the fact that the original graph does not contain any instance of the subgraphs of Lemmas 3.1 and 5.3, whenever u and v are consecutive backbone nodes with $\pi(u) < \pi(v)$ the sets $\kappa(R(u) \cup \{u\})$ and $\kappa(L(v) \cup \{v\})$ are disjoint. Consider the partial layout that places first u, then all nodes in $R(u)$ (whatever order), then all nodes in $L(u)$ (whatever order) and finally places v. As there are no conflicts between the different blocks, we can add all the decorations of the nodes in $R(u)$, $L(v)$, and v just after v, placing them in the same order as their corresponding nodes, the final layout is a proper colored layout. $\qquad\square$

From the previous theorem it is easy to obtain an algorithm that computes a proper colored layout when there is one. Furthermore the algorithm can be easily

implemented in a PRAM machine, using a polynomial number of processors and a polylogarithmic number of steps.

Theorem 5.5. PCLP *is in NC for decorated caterpillars with hair length 1, and* PICG *is in NC for caterpillars with hair length 1.*

References

[1] C. Àlvarez, J. Díaz and M. Serna, The hardness of intervalizing four colored caterpillars. *Discrete Math.* **235** (2001) 19–27.

[2] C. Àlvarez, J. Díaz and M. Serna, *Intervalizing colored graphs is NP-complete for caterpillars with hair length 2*. Technical Report LSI 98-9-R, Universitat Politècnica de Catalunya (1998).

[3] H. Bodlaender, M.R. Fellows and M.T. Hallet, Beyond NP-completeness for problems of bounded width: hardness for the W-hierarchy, in *26th ACM Symposium on Theory of Computing* (1994) 449–458.

[4] J. Díaz, A.M. Gibbons, M.S. Paterson and J. Torán, The minsumcut problem, in *Algorithms and Datastructure*, edited by F. Dehen, R.J. Sack and N. Santoro. *Lect. Notes Comput. Sci.* **519** (1991) 65–79.

[5] M.J. Dinneen, *VLSI Layouts and DNA physical mappings*. Technical Report, Los Alamos National Laboratory (1996).

[6] M.R. Fellows, M.T. Hallet and W.T. Wareham, DNA physical mapping: Three ways difficult, in *Algorithms-ESA'93*, edited by T. Lengauer. *Lect. Notes Comput. Sci.* **726** (1993) 157–168.

[7] P.W. Goldberg, M.C. Golumbic, H. Kaplan and R. Shamir, Four strikes against physical mapping of DNA. *J. Comput. Biol.* **2** (1995) 139–152.

[8] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco (1979).

[9] M.C. Golumbic, H. Kaplan and R. Shamir, On the complexity of DNA physical mapping. *Adv. Appl. Math.* **15** (1994) 203–215.

[10] M.C. Golumbic, H. Kaplan and R. Shamir, Graph sandwich problems. *J. Algorithms* **19** (1995) 449–473.

[11] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*. Academic Press, New York (1980).

[12] M.C. Golumbic and R. Shamir, Complexity and algorithms for reasoning about time: A graph theoretical approach. *J. ACM* **40** (1993) 1108–1113.

[13] D. Kuo and G.J. Chang, The profile minimization problem in trees. *SIAM J. Comput.* **23** (1994) 71–81.

[14] H. Kaplan and R. Shamir, Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques. *SIAM J. Comput.* **25** (1996) 540–561.

[15] H. Kaplan, R. Shamir and R.E. Tarjan, Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM J. Comput.* **28** (1999) 1906-1922.

[16] B. Monien, The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM J. Algebr. Discrete Methods* **7** (1986) 505–512.