Restrictions and preassignments in preemptive open shop scheduling

D. de Werra\textsuperscript{a,}\textsuperscript{*}, A.J. Hoffman\textsuperscript{b}, N.V.R. Mahadev\textsuperscript{c}, U.N. Peled\textsuperscript{d}

\textsuperscript{a}D\textsuperscript{é}partement de Math\textsuperscript{é}matiques, Ecole Polytechnique F\textsuperscript{é}d\textsuperscript{é}rale de Lausanne, CH-1015 Lausanne, Switzerland
\textsuperscript{b}Thomas J. Watson Research Center IBM, Yorktown Heights, New York, USA
\textsuperscript{c}Department of Mathematics, Northeastern University, Boston, USA
\textsuperscript{d}Department of Mathematics, Statistics \& Computer Science, University of Illinois at Chicago, USA

Received 24 February 1994; revised 27 January 1995

Abstract

Preemptive open shop scheduling can be viewed as an edge coloring problem in a bipartite multigraph. In some applications, restrictions of colors (in particular preassignments) are made for some edges. We give characterizations of graphs where some special preassignments can be embedded in a minimum coloring (number of colors = maximum degree). The case of restricted colorings of trees is shown to be solvable in polynomial time.

Keywords: Chromatic scheduling; Open shop scheduling; Timetabling; Production; Edge-coloring

1. Introduction

Among the classical models of scheduling, the open shop scheduling model has received much attention (see references in [3]). The reason is that such a model occurs in simplified formulations of many real scheduling problems. Class-teacher timetabling is one such problem where the (preemptive) open shop model is a natural basic formulation. However, in practice there are many additional requirements which have to be introduced in the open shop model in order to derive a solution which gives a timetable that can really be used in a school. Among these requirements are the so-called preassignments: some lectures have to be scheduled at periods which are fixed in advance. Can one construct a timetable in $k$ time units (periods) which satisfies these requirements?

\textsuperscript{*}Corresponding author.
In general, the class-teacher timetabling problem with preassignments is NP-complete [6]; our purpose is to consider some special types of preassignments and to derive necessary and sufficient conditions for the existence of such a timetable. These conditions will be expressed in terms of open shop scheduling. A graph-theoretical model will lead to characterizations of classes of graphs for which some extension properties of edge colorings are satisfied.

All graph-theoretical terms not defined here can be found in [1].

We first give a formulation of the open shop scheduling model and we will describe an associated edge coloring model which will be used consistently for handling the preassignment requirements.

We are given a set $\mathcal{P}$ of $m$ processors $P_1, \ldots, P_m$ and a collection $\mathcal{J}$ of $n$ jobs $J_1, \ldots, J_n$ to be processed within a period of $k$ consecutive time units. Each job $J_j$ consists of tasks $T_{1j}, \ldots, T_{mj}$; task $T_{ij}$ of job $J_j$ has to be processed on processor $P_i$; its processing time $p_{ij}$ is given and we assume that it is integral. If $p_{ij} = 0$ then $T_{ij}$ does not exist. No processor can work on two tasks simultaneously and no two tasks of the same job can be processed at the same time. The tasks of the same job can be processed in any order. We, furthermore, assume that preemptions are allowed (after any integral number of time units) during the processing of a task on a processor.

The question is whether it is possible to schedule all jobs within $k$ time units while satisfying all requirements described above. A classical application of this model is the simple class-teacher timetabling problem: each $P_i$ is a teacher and each $J_i$ is a class, i.e. a group of students taking exactly the same program. Then $T_{ij}$ is the collection of $p_{ij}$ lectures (of one time unit each) that teacher $P_i$ has to give to class $J_j$.

Associate with the problem a bipartite multigraph $G = (\mathcal{P}, \mathcal{J}, E)$ constructed as follows: each $P_i$ corresponds to a node in the left set $\mathcal{P}$ of nodes and each $J_j$ to a node in the right set $\mathcal{J}$ of nodes. Node $P_i$ is linked to node $J_j$ by $p_{ij}$ parallel edges. An edge $k$-coloring of $G$ is an assignment $F$ of one color $F(e) \in \{1, \ldots, k\}$ to each edge $e$ of $G$ such that $F(e) \neq F(g)$ whenever edges $e$ and $g$ are adjacent (i.e. share at least one node). Edge $k$-colorings of $G$ and feasible schedules in $k$ time units for the above timetabling problem or for the open shop scheduling problem are in correspondence: $F([P_i, J_j]) = q$ means that task $T_{ij}$ is in process during the $q$th time unit and that teacher $P_i$ and class $J_j$ meet at that time. If $\Delta(G)$ denotes the maximum degree of $G$ (the maximum number of edges incident to the same node), we know such an edge $k$-coloring exists if and only if $k \geq \Delta(G)$ (theorem of König, see [1]).

In the remainder of the paper we shall examine mainly situations where a subset $Q$ of edges has a preassigned color $q$ and another subset $R$ has a preassigned color $r \neq q$. We shall try to characterize the situations where this precoloring of $G$ can be extended to an edge $\Delta(G)$-coloring of the whole graph. Without loss of generality we can assume $q = 1$ and $r = 2$. Such precolorings correspond to preassignments of tasks to processors at some periods in the open shop model or to preassignments of some meetings in the timetabling formulation. In general, we may have more than two subsets of edges which have a preassigned color. The problem is generally NP-complete. Furthermore, it remains NP-complete when there are only two subsets $Q$,
R and even for the case $\Delta(G) = 3$ (see [6]); so we will concentrate on special sets $Q \cup R$ of precolored edges or on special graphs. One should at this stage observe that the problem of extending a partial edge coloring in a graph $G$ to an edge $k$-coloring of $G$ is a special case of restricted edge coloring [11]. Complexity issues are discussed in [11] where the case of trees is unknown. We shall deal with this case in the next section. Extensions to interval colorings are presented in [12].

First we will give a mathematical programming formulation of the problem and discuss complexity.

2. Restricted edge-colorings of trees

We now need to define restricted edge colorings which are closely related to precolorings. Let $G = (V, E)$ be a connected graph, $C$ a set of colors. For each $e \in E$, let $\varphi(e) \subseteq C$ be given and $\varphi(E) = \{\varphi(e): e \in E\}$. The restricted edge-coloring problem $(G, \varphi)$ is to find an edge coloring (i.e. a function $f: E \rightarrow C$ such that adjacent edges have distinct colors), with $f(e) \in \varphi(e)$ for all $e \in E$.

In fact, when some edges are precolored, we change the set $\varphi(e)$ of edges $e$ which are adjacent to some precolored edge. More precisely, we have initially $\varphi(e) = \{1, \ldots, k\}$ for each edge of the graph $G = (V, E)$. We examine consecutively all precolored edges: if $e$ has received color $i$ we set $\varphi(e) = \{i\}$ and remove $i$ from all sets $\varphi(f)$ of edges $f$ adjacent to $e$.

Let $\Gamma$ be a collection of pairs $(v, c)$, where $v \in V$, $c \in C$. We say that $\Gamma$ covers an edge $e = (v_1, v_2)$ of $(G, \varphi)$ if, for every $c \in \varphi(e)$, $(v_1, c) \in \Gamma$ or $(v_2, c) \in \Gamma$. Let $E(\Gamma)$ be the set of edges covered by $\Gamma$. Then, as we show later, for trees we have:

\[(G, \varphi) \text{ has no solution if and only if there is a } \Gamma \text{ with } |\Gamma| < |E(\Gamma)| \]  

(2.1)

For reasons which will be clear later, we will call a $\Gamma$ satisfying the assumptions of (2.1) a Hall certificate (of non-colorability) for $(G, \varphi)$. The principal aim of this section is to characterize those $G$ with the property that, for every $\varphi$, $(G, \varphi)$ has either a solution or a Hall certificate.

Let $A(G)$ be the node-edge incidence matrix of $G$ ($a_{ij} = 1$ if node $i$ is an endpoint of edge $j$ and $a_{ij} = 0$ else), let $k \geq \Delta(G)$ be the number of colors, let $B$ be the matrix given in Fig. 1. Recall that a $(0, 1)$ matrix is balanced if it does not contain a square $A$ and $I$ have each appeared $k$ times 

\[D(G) = \begin{bmatrix} A \\ & A \\ & & \ddots \\ & & & A \\ I & I & \ldots & I \end{bmatrix} \]

Fig. 1. The constraint matrix of the edge coloring problem.
submatrix of odd order such that each row and each column contains exactly two 1’s. The following result characterizes balanced matrices [2]:

**Theorem 2.0** (Fulkerson et al. [7]). For a \((0, 1)\) matrix \(M\) the following are equivalent:

(a) \(M\) is balanced;

(b) for every submatrix \(A\) of \(M\), the extreme points of the set-packing polytope \(\{x: Ax \leq 1, 0 \leq x \leq 1\}\) are integral;

(c) for every submatrix \(A\) of \(M\), the extreme points of the set-covering polytope \(\{x: Ax \geq 1, 0 \leq x \leq 1\}\) are integral.

**Theorem 2.1.** Let \(G\) be a connected graph. Then the following statements about \(G\) are equivalent:

(i) \(G\) is a tree;

(ii) \(B(G)\) is balanced for every \(k\);

(iii) For every \((G, \phi)\), there is either a solution or a Hall certificate.

**Proof.** We shall now prove that 2.1(i) implies 2.1(ii).

Essentially the proof will consist in considering a \(p \times p\) submatrix \(A'\) of \(B(G)\) with exactly two 1’s in each row and in each column. We will show that \(A'\) must be of even order, this will prove that \(B(G)\) is balanced.

\(B(G)\) consists essentially of \(k\) diagonal blocks \(A^1, \ldots, A^k\) (all identical to the node-edge incidence matrix \(A\) of the tree \(G\)) and of a horizontal band \(R\) of \(k\) identity \((m \times m)\)-matrices \(I^1, \ldots, I^k\). Let \(v_1, \ldots, v_m\) be the nodes of \(G\) and \(e_1, \ldots, e_m\) its edges (we clearly have \(m = n - 1\)).

We may consider that we have \(k\) copies \(G^1, \ldots, G^k\) of the tree \(G\) on node sets \(\{v_1^1, \ldots, v_m^1\}, \ldots, \{v_1^k, \ldots, v_m^k\}\) respectively and with edge sets \(\{e_1^1, \ldots, e_m^1\}, \ldots, \{e_1^k, \ldots, e_m^k\}\), respectively. Furthermore, for each edge \(e_i\) of \(G\) there is a row \(\tilde{e}_i\) having 1’s in columns associated to \(e_i^1, e_i^2, \ldots, e_i^k\).

Now for each column \(e_i^r\) of \(A'\) its two 1’s are either in the same block \(A'\) (in which case they are in the rows \(v_a^r, v_b^r\) associated to the endnodes \(v_a^r, v_b^r\) of edge \(e_i^r\) in \(G^r\)), i.e. corresponding to the endnodes \(v_a, v_b\) of edge \(e_i\) in \(G\), or there is a 1 in the row \(v_a^r\) and a 1 in row \(\tilde{e}_i\) of \(I^r\) in the band \(R\).

For each row of \(A'\) we have the following:

(a) If the row is \(v_a^r\) in \(A'\) (i.e. it corresponds to node \(v_a^r\) of \(G^r\)), the two 1’s are in columns \(e_i^r, e_j^r\) where \(e_i, e_j\) are two edges of \(G\) which are adjacent to node \(v_a\).

(b) If the row is \(\tilde{e}_i\) in band \(R\), the two 1’s are in columns \(e_i^r, e_i^s\) corresponding to the same edge in \(G^r\) and in \(G^s\) (where \(r, s\) are distinct arbitrary integers between 1 and \(k\)).

Let us assume without loss of generality that \(A'\) defines a cycle which is obtained by starting from any 1 in \(A'\), moving alternately to the other 1 in the row and to the other 1 in the column and coming back to the starting 1. This walk can be interpreted in the following way:
**Horizontal move:** (from a 1 to the other 1 in the same row). If we are in row \( v'_0 \) or \( A' \) and move from entry \( (v'_0, e'_1) \) to entry \( (v'_0, e'_s) \), this corresponds to moving from edge \( e'_1 \) (adjacent to \( v'_0 \)) to edge \( e'_s \) (also adjacent to \( v'_0 \)) in \( G' \).

If we are in row \( e_i \) of band \( R \), a move from entry \( (e_i, e'_1) \) to entry \( (e_i, e'_s) \), corresponds to moving from edge \( e'_1 \) of \( G' \) to the corresponding edge \( e'_s \) of \( G^k \) \( (s \neq r) \). So either we move from one edge of some \( G' \) to an adjacent edge of the same \( G' \), or we move from one edge of some \( G^k \) to the corresponding edge of some \( G^k \) with \( s \neq r \).

**Vertical move:** (from a 1 to the other 1 in the same column). Suppose we are in column \( e'_r \) corresponding to an edge of \( G' \) with endnodes \( v'_u, v'_b \). A move from \( (v'_u, e'_j) \) to \( (v'_u, e'_s) \) corresponds to moving from one endnode \( v'_u \) of \( e'_j \) to the other endnode \( v'_u \) of this edge.

Another type of move is from entry \( (v'_u, e'_j) \) in \( A' \) to entry \( (e_i, e'_s) \) in band \( R \) (or conversely). This corresponds to moving from node \( v'_u \) of \( G' \) to some "artificial" node \( e_i \) of \( e'_s \); we may indeed consider that the rows in \( R \) correspond to some artificial nodes. These artificial nodes allow us to go from some edge to the corresponding edge in another graph.

Now we start from a 1, say in entry \( (v'_u, e'_j) \) and follow the "cycle" defined by \( A' \) as described. As long as we remain in \( A' \), we follow an elementary chain in \( G' \) starting at \( v'_u \). When we leave \( A' \) it must be by a vertical move from, say, \( (v'_u, e'_j) \) to \( (e_i, e'_s) \) in \( R \) and this entry is necessarily followed by some entry \( (e_i, e'_s) \) with \( s \neq r \), which is again necessarily followed by a vertical move to the entry \( (v'_u, e'_j) \) or \( (e_i, e'_s) \) where \( v'_b, v'_c \) are the ends of the edge \( e_j \). In fact, the move is to entry \( (v'_u, e'_j) \) and not to \( (e_i, e'_s) \) for the following reason. If the move were to the entry \( (v'_u, e'_j) \), then, \( G \) being a tree, in order for the cycle to return to the starting entry it must use the edge \( e_j \) in a different block (to reach \( v'_b \)). But then the row of \( A' \) corresponding to \( e_j \) will have three ones, a contradiction. Now we follow an elementary chain in \( G^k \) starting at \( v'_b \). So if we identify all graphs \( G^1, \ldots, G^k \) with \( G \), and consider the projection of the cycle in \( G \), we observe that the vertical moves from \( G' \) to \( R \) and from \( R \) to \( G^k \) can be thought of as backtracking along the edge \( e_j \) to its end \( v'_b \). Thus every vertical move corresponds to travelling from one end to the other end of an edge, while every horizontal move corresponds to selecting the next adjacent edge to travel. We move along a chain of \( G \) starting at \( v_u \) and coming back to \( v_u \). Since \( G \) is a tree, this requires that every edge is traveled an even number of times. Thus \( p \), the number of vertical moves, is even. This ends the proof.

Next, we prove 2.1(ii) implies 2.1(iii). The blocks of \( B(G) \) are each naturally associated with colors. Delete all columns \( (c, e) \) where \( c \notin \varphi(e) \). The matrix which is left is denoted by \( B(G, \varphi) \).

Consider now the linear programming problem: \( \max 1' \cdot x : B(G, \varphi), x \leq 1, x \geq 0 \). As a submatrix of \( B(G) \), the matrix \( B(G, \varphi) \) is balanced, so the maximum is attained by Theorem 2.0 at a \( (0, 1) \) vector \( x \), which assigns at most one color to each edge. Each column of \( B(G, \varphi) \) corresponds to a pair \( (e, c) \) where \( e \) is an edge of \( G \) and \( c \) a color in \( \varphi(e) \). In the solution \( x(e, c) = 1 \) if edge \( e \) gets color \( c \) and \( x(e, c) = 0 \), otherwise. The maximum is \( |E| \) (i.e. every edge is assigned exactly one color) if and only if \( (G, \varphi) \) has a solution. Suppose the maximum is smaller than \( |E| \). Then by the duality theorem of linear programming and Theorem 2.0, the dual problem
min \(y : y' B(G, \phi) \geq 1', y \geq 0\) has an optimum \(\bar{y}\) which is \((0, 1)\) and achieves a value smaller than \(|E|\). The dual constraints are of the following form for each edge \(e = [v_1, v_2]\) and for each color \(c \in \varphi(e)\):

\[
y(v_1, c) + y(v_2, c) + y(e) \geq 1.
\]

The dual objective function is \(\sum_c \sum_v y(v, c) \leq \sum_v y(e)\). Let \(\Gamma = \{(v, c) : \bar{y}(v, c) = 1\}\) and \(F = \{e : e \in E \text{ with } \bar{y}(e) = 0\}\). From the dual constraints we have \(F \subseteq E(\Gamma)\), and from the dual objective function value \(|\Gamma| + |E - F| < |E|\) so \(|\Gamma| < |F| \leq |E(\Gamma)|\). Therefore, \(\Gamma\) is a Hall certificate. The converse of this argument also proves (2.1).

To prove 2.1(iii) implies 2.1(i) let \(G\) have a cycle \((u_0, v_1), (u_1, u_2), \ldots, (u_{r-1}, u_r), (u_r, u_0)\). Assign to each remaining edge \(e\) a different color \(c(e) \in \{1, \ldots, \rho\}\). Let \(\varphi(u_0, v_1) = \{1\}, \varphi(v_1, v_2) = \{1, 2\}, \ldots, \varphi(u_r, v_0) = \{r - 1, r\}, \varphi(v_r, u_0) = \{r, 1\}\). Then there is no solution to \((G, \varphi)\), as can be seen easily. Furthermore, there is no Hall certificate: due to the choice of colors for the edges which are not in the cycle \(D\), if there is a certificate \(\Gamma\) there is one with \(E(\Gamma)\) contained in the set \(E(D)\) of edges of the cycle. For any proper subset of \(E(D)\) the coloring problem has a solution, hence there is no certificate with \(|E(\Gamma)| < |E(D)|\). Now to cover all edges of the cycle we need at least one pair \((v, c)\) for each color \(c = 2, \ldots, \rho\) and at least two pairs for \(c = 1\) to cover edges \((v_r, v_0), (v_0, v_1), (v_1, v_2)\). So there is no certificate.

Since \(B(G, \varphi)\) is a balanced matrix, it follows from the fact that linear programming problems can be solved in polynomial time (see [9]), that a solution to \((G, \varphi)\) or a Hall certificate can be found in polynomial time. A polynomial algorithm based on the above discussion would have complexity dominated by the time to solve the linear program \(\max 1 \cdot x \cdot B(G, \varphi) \cdot x \leq 1, x \geq 0\). Whereas the complexity of the ellipsoid algorithm is quite high, there could be faster algorithms for linear programming with a balanced \((0, 1)\) coefficient matrix. However, we exhibit below a more efficient combinatorial algorithm tailored for this particular problem.

We will start by giving a combinatorial algorithm for restricted edge coloring of trees, due to Don Coppersmith of the T.J. Watson Research Center, IBM. The algorithm requires an introduction that will be detailed below. The principal step is best described by first considering the case where the tree \(T\) is a star with center \(w\). Think of the edges (which all have \(w\) as an endpoint) as sets of colors: i.e. edge \(e\) contains as “elements” the colors in \(\varphi(e)\). To solve \((T, \varphi)\) is to find a system of distinct representatives for the sets, and network flow theory (cf. [1] or many other sources) will produce the coloring or a Hall certificate \(\Gamma = \{(w, c_1), \ldots, (w, c_q)\}\) which covers more than \(q\) edges.

Next, assume that \(T\) is a general tree, and consider \(T\) as rooted at some node of degree 1. The root edge is the edge incident to the root node. We use the usual terminology that an edge \(f\) is a child of an edge \(e\) if \(e\) is the immediate predecessor of \(f\) on the unique chain from the root edge to \(f\). The descendants of \(e\) are its children and their descendants. By climbing \(T\) up from the leaves, we shall partition each \(\varphi(e)\) into a set \(\alpha(e)\) of “good colors” and a set \(\beta(e)\) of “bad colors” so that the following two conditions are satisfied.
For every good color \( c \in \alpha(e) \), the problem \((L, \sigma)\) has a feasible coloring, where \( L \) is the star consisting of \( e \) and its children, \( \sigma(e) = \{c\} \) and \( \sigma(f) = \alpha(f) \) for every child \( f \) of \( e \).

The problem \((S, \psi)\) has a Hall certificate \( \Gamma(e) \), where \( S \) is the subtree consisting of \( e \) and its descendants, \( \psi(e) = \beta(e) \), and \( \psi(f) = \varphi(f) \) for every descendant \( f \) of \( e \).

We define \( \alpha(e) \) as the set of those colors in \( \varphi(e) \) that satisfy the first condition, and \( \beta(e) = \varphi(e) - \alpha(e) \). We then have to show how to obtain the certificate \( \Gamma(e) \) in the second condition. We shall do this in a moment, but let us point out a consequence of this fact. If any edge \( e \) satisfies \( \alpha(e) = \emptyset \) and therefore \( \beta(e) = \varphi(e) \), then \( \psi \) is \( \varphi \) restricted to the subtree \( S \), hence \( \Gamma(e) \) is in fact a certificate for the original problem \( (T, \varphi) \) and we are done. On the other hand, if we have reached the root edge \( e \) without stopping on the way and find \( \alpha(e) \neq \emptyset \), then we can choose for \( e \) and all its children distinct good colors. Hence by the first condition we can also choose good colors for the grandchildren of \( e \) consistent with the above choice, and continuing in this way we find a feasible coloring for \( (T, \varphi) \).

It remains to show how to construct the certificate \( \Gamma(e) \) in the second condition. By definition of \( \beta(e) \), for each \( c \in \beta(e) \), \((L, \sigma)\) has no feasible coloring. It follows that \((L, x)\) has no feasible coloring either, where \( x(e) = \beta(e) \) and \( x(f) = \alpha(f) \) for every child \( f \) of \( e \). Since this is a coloring problem on a star \( L \), we can construct a Hall certificate \( \Gamma'(e) \) for it by network flow techniques, as indicated earlier. It satisfies

\[
|\Gamma'(e)| \leq |E(\Gamma'(e))| - 1. \tag{2.2}
\]

Recursively, we have already constructed the certificates \( \Gamma(f) \) for all children \( f \) of \( e \), if any. They satisfy

\[
|\Gamma(f)| \leq |E(\Gamma(f))| - 1
\]

By summing these inequalities over all children \( f \) of \( e \) and using the fact that the \( E(\Gamma(f)) \) are disjoint sets of edges (since they are contained in edge-disjoint subtrees), we obtain

\[
\left| \bigcup_f \Gamma(f) \right| \leq \left| \bigcup_f E(\Gamma(f)) \right| - p, \tag{2.3}
\]

where \( p \) is the number of children of \( e \). We construct the required certificate \( \Gamma(e) \) by

\[
\Gamma(e) = \Gamma'(e) \cup \bigcup_f \Gamma(f).
\]

To show that \( \Gamma(e) \) is indeed a Hall certificate for \( (S, \psi) \), we distinguish two cases. If \( e \notin E(\Gamma'(e)) \), then

\[
|E(\Gamma'(e))| \leq p \tag{2.4}
\]

and hence by summing inequalities (2.2), (2.3) and (2.4), we obtain

\[
|\Gamma(e)| \leq \left| \bigcup_f E(\Gamma(f)) \right| - 1 \leq |E(\Gamma(e))| - 1. \tag{2.5}
\]
as required. On the other hand, if \( e \in E(I'(e)) \), we lose 1 in (2.4), but recover it in the second inequality sign of (2.5). The reason is that \( \bigcup_f I'(f) \) does not cover \( e \) (since it is included in the set of descendants of \( e \)), but \( I''(e) \) does, and hence \( \bigcup_f I'(f) \) is a proper subset of \( E(I(e)) \).

Let us now study the complexity of the above algorithm. If the tree \( T \) is a star, we solve a System of Distinct Representatives (SDR) problem where the sets are the \( \varphi(e) \) for the edges \( e \) of the star. This is a maximum matching problem in a bipartite graph where nodes on one side correspond to the edges of the star and nodes on the other side correspond to the colors. If \( A \) is the maximum degree of the tree and \( k \) is the total number of colors available, the bipartite matching problem involves at most \( A + k \) nodes. According to [10] we can color the edges of the star or get a Hall certificate in time \( O(A + k)^{2.5} \).

In Coppersmith's algorithm, if the tree \( T \) is not a star, we root \( T \) on a node of degree 1. For each edge \( e \) that is not a leaf edge, we partition \( \varphi(e) \) into \( \alpha(e) \) and \( \beta(e) \). To do so, we consider the star \( L \) consisting of \( e \) and its children edges. To each child \( f \) we give \( \alpha(f) \) as the set of available colors, and to \( e \) we give a single color \( c \in \varphi(e) \) as available. We solve the corresponding SDR problem. If there is a SDR, \( c \) goes to \( \alpha(e) \), and if not, \( c \) goes to \( \beta(e) \). So we solve \( |\varphi(e)| \) SDR problems for \( e \). Finally, we solve one more SDR problem for \( e \), where \( e \) is given the whole set \( \beta(e) \) as available colors, in order to get the certificate \( I''(e) \). So altogether, the work on \( e \) is done in time \( (1 + |\varphi(e)|) O(A + k)^{2.5} = O(k(A + k)^{2.5}) \). The overall work is \( O(n'(A + k)^{2.5}) \) where \( n' \) is the number of edges with children, i.e. \( n' = 1 + \) no. on nodes of degree greater than 1 (actually for the root edge we need just one SDR problem).

From now on we will concentrate on precolorings which are, as shown at the beginning of the section, special cases of restricted colorings. We shall describe some special cases where precolorings in trees can be extended by simple graph-theoretic algorithms based on exchange chains. It will be convenient to consider multigraphs as being obtained from simple graphs \( G \) by multiplying each edge \( e \) by a nonnegative weight \( w_e \); the resulting graph will be a "weighted" graph \( G_w \): \( w_e = 0 \) means that the edge \( e \) is deleted and \( w_e \geq 2 \) means that \( e \) is replaced by \( w_e \) parallel edges.

We shall first consider the case where \( G = (V, E) \) is a tree.

**Fact 2.1.** Let \( G \) be a tree; assume the set \( D \) of precolored edges is connected and does not involve more than \( A(G) \) colors. Then the coloring of \( D \) can be extended to an edge \( A(G) \)-coloring.

**Proof.** The coloring can be extended from \( D \) by coloring the edges around each node consecutively while considering the nodes in the order of their distance to \( D \). At each node where an uncolored edge is present, there is some color \( i \leq A(G) \) which is missing. An edge \( A(G) \)-coloring will be obtained in this way. \( \square \)

Notice that Fact 2.1 does not hold when \( G \) is not a tree (see Fig. 2).

According to the observation made in Section 1 we shall restrict our attention to precolorings consisting of a subset of edges which have received color 1 and color 2.
Such subsets can hence be only unions of node-disjoint chains and even cycles. So let
us assume that edge-disjoint matchings $Q$ and $R$ are given (edges in $Q$ must have color
1 and edges in $R$ must have color 2). $Q \cup R$ is a collection of node disjoint chains if the
graph is a tree.

**Fact 2.2.** If $G$ is a tree and $Q \cup R$ has at most two connected components, then the
coloring of $Q \cup R$ can be extended to an edge $k$-coloring with $k \leq \Delta(G) + 1$.

**Proof.** Remove one edge $e$ of $G$ so that $G$ is disconnected into two connected
components, each one of them containing at most one connected component of $Q \cup R$.
Clearly in $G-e$ the coloring of $Q \cup R$ can be extended to an edge $\Delta(G)$-coloring from
Fact 2.1. Then give $e$ color $\Delta(G) + 1$. □

In light of Facts 2.1 and 2.2, it is natural to ask the following question: if $G$ is a tree
and $Q \cup R$ has two connected components, under what conditions can the precoloring
of $Q \cup R$ be extended to an edge $\Delta(G)$-coloring of $G$? To answer this question we need
the following definition.

In a multigraph $G_w = (V, E)$ a **$p$-matching** is a subset $F$ of edges such that each node
is adjacent to at most $p$ edges of $F$; $f(z)$ will be the number of edges of $F$ which are
adjacent to node $z$. A $p$-matching $F$ will be called **admissible** in $G_w$ if
$\Delta(G_w - F) = \Delta(G_w) - p$.

In a bipartite multigraph $G_w$, the edges of a $p$-matching $F$ can be colored with
$p$ colors. If $F$ is admissible, the remaining edges (i.e. the edges in $E - F$) can be colored
with $\Delta(G_w) - p$ colors. Both facts follow from the theorem of König.

**Remark 2.1.** Finding an admissible $p$-matching in a bipartite multigraph $G$ can be
done in polynomial time, since it is a compatible flow problem. In particular it can be
done for trees.
Proposition 2.3. Let $G$ be a tree with $\Delta(G) \geq 3$. Assume $Q \cup R$ has two connected components. Then there exists an edge $\Delta(G)$-coloring of $G$ extending the coloring of $Q \cup R$ if and only if there exists an admissible 2-matching $F$ with $F \supseteq Q \cup R$.

Proof. The “only if” part is immediate: if $Q, R$ are given, it is necessary for an edge $\Delta(G)$-coloring extending the coloring of $Q \cup R$ to exist that there exists an admissible 2-matching $F$ which contains $Q \cup R$; $F$ will consist of all edges which will get color 1 or color 2. Let us now assume that there exists an admissible 2-matching $F$ which contains $Q \cup R$. Let $S_a, S_b$ be the connected components of $Q \cup R$. If $S_a$ and $S_b$ are not in the same connected component of $F$, then by Fact 2.1 we may bicolor separately the connected components of $F$ while giving color 1 (resp. color 2) to the edges in $Q$ (resp. $R$).

Now assume $S_a, S_b$ (which are chains) are in the same connected component of $F$ (which is also a chain). Let $[x, y]$ be the first edge of the unique chain in $F - (Q \cup R)$ joining $S_a$ to $S_b$. We will remove it from $F$ in the procedure described below. Notice that such an edge always exists. If $\max(d(x), d(y)) \leq \Delta(G) - 1$ where $d(z)$ is the degree of node $z$, then we may remove $[x, y]$ from $F$; the remaining 2-matching is still admissible and $S_a, S_b$ are now in different components of the 2-matching and we are in the previous case.

If $d(x) = \Delta(G)$, then we label $[x, y]$ with $\ominus$ and since $\Delta(G) \geq 3$ there is an edge $[x, z] \not\in F$; we label it with $\ominus$; we will now construct an (elementary) alternating chain which will be used to modify $F$. For this purpose we construct an alternating sequence of edges in $F$ and not in $F$ (labelled with $\ominus$ and with $\ominus$) starting at $x$ with $[x, z]$. We extend the chain as far as possible from $x$. We stop when we reach with a $\ominus$-edge a node $v$ with $f(v) < 2$ or with a $\ominus$-edge a node $u$ with either $f(u) = 2, d(u) \leq \Delta(G) - 1$ or $f(u) = 1, d(u) \leq \Delta(G) - 2$. This should happen since $G$ is a tree: the chain $C$ constructed is elementary (no node occurs more than once). So if we do not stop before, we will reach a node $v$ with $d(v) = 1$ and hence we cannot continue from $v$. Let $[u, v]$ be the last edge of the chain. Notice that $d(v) < \Delta(G)$ otherwise $d(v) = \Delta(G) = 1$ and $G$ has only nodes $u$ and $v$, contrary to the assumptions. We have two cases to consider:

(a) If $[u, v]$ is a $\ominus$-edge, then $f(v) \leq d(v) < 2$ and so we stop at $v$ according to the rule.

(b) If $[u, v]$ is a $\ominus$-edge, then in case $f(v) = 1$ and $d(u) \leq \Delta(G) - 2$ we also stop according to the rule. Suppose now that $f(v) = 1$ and $d(v) = \Delta(G) - 1$. So we have $\Delta(G) = 2$, contrary to the assumption; so this case is not possible.

So the chain $C$ will stop according to the rule in all cases. The chain cannot reach any component of $Q \cup R$ since $G$ has no cycles. If $d(y) = \Delta(G)$ we similarly extend the chain from $y$ by starting with some edge $[y, s] \not\in F$. It will be disjoint from the part of the chain grown from $x$. By dropping from $F$ all the $\ominus$-edges and adding all the $\ominus$-edges we get an admissible 2-matching $F'$ such that $F' \supseteq Q \cup R$ and $S_a, S_b$ are in different connected components of $F'$ and we are again in the first case. □

Observe that if there are more than two connected components, the construction of such an $F'$ may not be possible (see the example in Fig. 3 where $Q \cup R$ has 3 connected
components; for this example there exists however an admissible 2-matching $F$ containing $Q \cup R$.

\textbf{Remark 2.2.} We have assumed $\Delta(G_w) \geq 3$ in the previous statements. The case $\Delta(G_w) = 2$ is immediate for bipartite multigraphs. Each connected component of $G_w$ is either a cycle (of even length $\geq 2$) or a chain. A 2-coloring exists if and only if all edges in $Q$ are odd numbered edges and all edges in $R$ are even numbered edges or vice versa when numbering the edges consecutively along the chain or the cycle. Otherwise an edge 3-coloring can always be found. One should furthermore notice that it is not necessary in general that the connected components of $Q \cup R$ be in different connected components of the 2-matching $F$. However, this is sufficient for the existence of a bicoloring of the edges of $F$ taking the preassignment constraints into account. It turned out that for trees this condition could always be satisfied.

\textbf{Remark 2.3.} Clearly if the preassignment requirements consist of $Q, R$ then for any bipartite $G_w$ the minimum number of colors will never exceed $\Delta(G_w) + 2$, since we may get an edge $(\Delta(G_w) + 2)$-coloring by setting $M_1 = Q, M_2 = R$ and by coloring $G - (Q \cup R)$ with colors $3, 4, \ldots, \Delta(G_w) + 2$ (from the theorem of König). We assumed that $Q \cup R$ had at most two connected components. In fact if the number of connected components of $Q \cup R$ is strictly smaller than $\Delta(G_w)$, then for trees an almost admissible 2-matching $F^*$ (i.e. a 2-matching $F^*$ such that $\Delta(G_w - F^*) \leq \Delta(G_w) - 1$) can be found with $F^* \supseteq Q \cup R$. Then we can show that if the number of connected components of $Q \cup R$ is smaller than the maximum degree, there is an edge $(\Delta(G_w) + 1)$-coloring or an edge $\Delta(G_w)$-coloring satisfying the preassignment constraints.

In the case where $Q \cup R$ has one connected component, then we may state the following result for arbitrary bipartite multigraphs.

\textbf{Proposition 2.4.} Let $G_w$ be a bipartite multigraph, $Q, R$ subsets of edges which have to get color 1 and color 2, respectively. Assume $Q \cup R$ is connected. Then the minimum
number $k$ of colors in an edge $k$-coloring satisfying the requirements $Q, R$ can be found in polynomial time.

Proof. Now there exists an edge $\Delta(G_w)$-coloring of $G$ satisfying requirements $Q, R$ if and only if there is an admissible 2-matching $F^*$ with $F^* \supseteq Q \cup R$. Since $Q \cup R$ is connected, $F^*$ will be partitionable into $M_1, M_2$ with $M_1 \supseteq Q, M_2 \supseteq R$. If it exists, an admissible 2-matching $F^*$ containing $Q \cup R$ can be obtained as follows: remove all edges of $Q \cup R$ from $G$. To each node $z$ assign an integer $a(z)$ defined by

$$a(z) =
\begin{cases}
1 & \text{if } z \text{ was adjacent to one edge of } Q \cup R \\
2 & \text{if } z \text{ was adjacent to no edge of } Q \cup R \\
0 & \text{if } z \text{ was adjacent to two edges of } Q \cup R.
\end{cases}$$

Then we have to find a subset $F$ of edges in $G_w - (Q \cup R)$ such that for each node $z$

$$f(z) = a(z) \quad \text{if } d(z) = \Delta(G_w) \text{ in } G_w,$$

$$a(z) \geq f(z) \geq \max(0, a(z) - 1) \quad \text{if } d(z) = \Delta(G_w) - 1 \text{ in } G_w,$$

$$a(z) \geq f(z) \geq 0 \quad \text{otherwise.}$$

The above requirements will force $F$ to be a 2-matching such that $F \cup (Q \cup R)$ is still a 2-matching and each node $z$ with $d(z) = \Delta(G_w)$ (resp. $d(z) = \Delta(G_w) - 1$) is adjacent to exactly two edges (resp. at least one and at most two edges) of $F \cup (Q \cup R)$. Since $G_w$ is bipartite, this is a flow problem. If such an $F$ exists, then we can find an edge $\Delta(G_w)$-coloring. If it does not exist, we have to look for an almost admissible 2-matching $F$, i.e. a 2-matching $F$ such that $\Delta(G_w - F) \leq \Delta(G_w) - 1$. We define $a(z)$ as before, but now for each node $z$ we impose

$$a(z) \geq f(z) \geq \max(0, a(z) - 1) \quad \text{if } d(z) = \Delta(G_w) \text{ in } G_w,$$

$$a(z) \geq f(z) \geq 0 \quad \text{otherwise.}$$

This is again a compatible flow problem. If such a 2-matching $F$ exists, then $F \cup Q \cup R$ will have at least one edge adjacent to each node $z$ with $d(z) = \Delta(G_w)$. So it is almost admissible. Coloring $F \cup Q \cup R$ with colors 1 and 2 can be done while taking the requirements $Q, R$ into account. $G - (F \cup Q \cup R)$ can then be colored with $\Delta(G_w) - 1$ colors. We get an edge $(\Delta(G_w) + 1)$-coloring. If no such $F$ can be found, then we know from Remark 2.3 that the smallest number of colors needed is $\Delta(G_w) + 2$. 

Remark 2.4. If instead of $Q, R$ we have three subsets $Q, R, S$ of edges which must receive colors 1, 2 and 3, respectively, the above result may no longer be true: in the graph of Fig. 4, there exists an admissible 3-matching $F$ containing the set $Q \cup R \cup S$ of precolored edges (take the whole graph for $F$). Nevertheless, there is no edge $\Delta(G_w)$-coloring $(\Delta(G_w) = 3)$ satisfying the preassignment requirements.
3. Other special cases of precolorings

We shall now consider again the case where preassignments consist of two subsets \( Q, R \) of edges which must have color 1 and color 2 respectively. Our purpose is to characterize classes of multigraphs \( G_w \) for which the precolorings can be extended to edge \( \Delta(G_w) \)-colorings.

A simple graph \( G \) has property \( EP(k) \) (extension of chain on \( k \) edges) if for any choice of weights \( w_e \) and for any chain \( P \) of at most \( k \) edges in \( G_w \), any bicoloring of the edges of \( P \) can be extended to an edge \( \Delta(G_w) \)-coloring of \( G_w \). Here our chains will be simple (all nodes are distinct).

**Proposition 3.1.** For a simple connected graph \( G \), the following statements are equivalent:

1. \( G \) satisfies \( EP(k) \) for all \( k \geq 3 \);
2. \( G \) satisfies \( EP(k) \) for some \( k, 3 < k < d \), where \( d \) is the length of the longest chain in \( G \);
3. \( G \) satisfies \( EP(3) \);
4. \( G \) is an even cycle or a tree.

**Proof.** Trivially (1) \( \Rightarrow \) (2) \( \Rightarrow \) (3). Also it is easy to verify that (4) \( \Rightarrow \) (1): if \( G_w \) is an even cycle (possibly with multiple edges), let \( P \) be a chain of length at least 3 with colors 1 and 2 alternating on its edges. One can extend the coloring with colors 1, 2 to a cycle \( C \). After removal of the edges of \( C \), the remaining graph can be colored with colors 3, 4, \ldots, \( \Delta(G_w) \).

If \( G_w \) is a multigraph obtained from a tree, the proof of Fact 2.1 applies.

Let us now show that (3) \( \Rightarrow \) (4). Suppose \( G \) is a connected graph which is neither an even cycle nor a tree. If \( G \) is bipartite, it must contain, as a partial subgraph, an even cycle with a pendent edge \([v_0, v_1]\) (see Fig. 5). Assign color 1 to \([v_0, v_1]\) and \([v_2, v_3]\) while \([v_1, v_2]\) gets color 2. Then give weight 2 to \([v_3, v_4], [v_5, v_6], \ldots, [v_{2q-1}, v_{2q}]\) while the other edges of the cycle and edge \([v_0, v_1]\) get weight 1. Clearly such a coloring cannot be extended to an edge 3-coloring, so \( G \) does not satisfy \( EP(3) \), which is a contradiction. If \( G \) is not bipartite, it must contain an odd cycle \( C \); let \( v_1, v_2, \ldots, v_{2q+1} \) be its nodes. If \( 2q + 1 \geq 5 \), let \([v_1, v_2], [v_3, v_4]\) get color 1 and \([v_2, v_3]\) get color 2. Give weight 2 to \([v_4, v_5], [v_6, v_7], \ldots, [v_{2q}, v_{2q+1}]\) and weight 1 to all remaining edges of \( C \) (see Fig. 6). Such a coloring cannot be extended to an edge 3-coloring of the currently weighted graph, so \( G \) does not satisfy \( EP(3) \). If \( 2q + 1 = 3 \),
$G$ contains a triangle; let $v_1, v_2, v_3$ be its nodes. Since $G$ satisfies $EP(3)$, it is not a triangle; so suppose there is an edge $[v_0, v_3]$ in $G$ with $v_2 \neq v_0 \neq v_3$. Give weight 2 to $[v_2, v_3]$ and weight 1 to $[v_0, v_1], [v_1, v_2], [v_1, v_3]$. Give color 1 to $[v_0, v_1]$ and to one edge $[v_2, v_3]$, while $[v_1, v_2]$ gets color 2. Clearly this coloring cannot be extended to an edge 3-coloring of the currently weighted graph and $G$ does not satisfy $EP(3)$, a contradiction.
Let us now consider another type of pressignment of colors: instead of chains we will consider cycles. We will therefore assume that all cycles are even. A simple graph $G$ satisfies property $EC$ (extendable cycle) if, for any choice of weights $w_e$ and for any cycle $c$ in $G_w$, any bicoloring of the edges of $c$ can be extended to an edge $\Delta(G_w)$-coloring of $G_w$. Furthermore let us call $EC_2$ (resp. $EC_1$) the property $EC$ where weights $w_e$ are restricted to $\{0, 1, 2\}$ (resp. to $\{0, 1\}$).

A *mouth* in a graph $G$ consists of three chains of the same parity which have the same endpoints but no intermediate node in common. The mouth is even (resp. odd) if the three chains are even (resp. odd) (see Fig. 7).

**Proposition 3.2.** For a simple bipartite graph $G$, the following statements are equivalent:

1. $G$ satisfies $EC$;
2. $G$ satisfies $EC_2$;
3. $G$ contains no even mouth as a partial subgraph.

**Proof.** Trivially we have (1) $\Rightarrow$ (2). Also (2) $\Rightarrow$ (3): let $G$ be an even mouth with three even chains $C_1$, $C_2$, $C_3$ with endpoints $x$, $y$. $C_1$ and $C_2$ form an even cycle $C$; we give weight 1 to all edges of $C$ and we color them alternately with colors 1 and 2. Let $[x, v_1]$, $[v_1, v_2]$, ..., $[v_{2q-2}, v_{2q-1}]$, $[v_{2q-1}, y]$ be the edges of $C_3$. Give weight 1 to $[x, v_1]$, $[v_2, v_3]$, ..., $[v_{2q-2}, v_{2q-1}]$, $[v_{2q-1}, y]$ and (if $q > 2$) weight 2 to $[v_1, v_2]$, $[v_3, v_4]$, ..., $[v_{2q-3}, v_{2q-2}]$ (see Fig. 8). It is easy to see that the bicoloring of $C$ cannot be extended to an edge 3-coloring of $G_w$, so $G$ does not have $EC_2$.

We should now show that (3) $\Rightarrow$ (1). We recall some properties of bipartite graphs containing no even mouths as partial subgraphs which are proved in [5]. These graphs are called BOC graphs (bipartite odd cactus). They are also characterized by a property of coloring which implies that if we choose one cycle $C$ in $G_w$ and bicolor its edges with colors 1 and 2, the coloring can be extended to an edge $\Delta(G_w)$-coloring. □
Remark 3.1. A recognition algorithm of BOC graphs $G = (V, E)$ in $O(|E|^3)$-time is also given in [5].

4. Coneless graphs and preassignments

In the previous section a property EC of precolorings in an open shop scheduling model has been studied; it led to the characterization of the class of simple bipartite graphs for which EC holds. In terms of open shop scheduling, such a property can also be viewed as follows: the data of an open shop scheduling problem consist of an $m \times n$ array $(m = |\mathcal{P}|, n = |\mathcal{F}|)$ containing the values $p_{ij}$. Each entry $[i, j]$ corresponds to an edge (or a family of parallel edges) of the associated graph $G$. In fact we can say that we characterized configurations $\mathcal{C}$ of entries such that whatever nonnegative integral values $w_e$ we introduce in the cells of $\mathcal{C}$ and whatever cycle we precolor with colors 1 and 2, an extension of a complete schedule in $A(G, w)$-time units can be found. ($A(G, w)$ is the maximum of all row sums and all columns sums in the array $(p_{ij})$.

Property EC is indeed a very strong requirement and leads to a restricted class of graphs; we may weaken the statement in EC and consider only weights $w_e \in \{0, 1\}$. We get property EC1. All graphs considered will now be simple.

A cone is an even mouth where at least one of the three chains has length two (Fig. 9). A simple graph is coneless if it contains no cone as a partial subgraph. An ear decomposition of a two-connected graph consists in repeatedly removing the intermediate nodes and the edges of inclusionwise maximal chains $\Pi_i$ whose intermediate nodes have degree two in the current graph until a cycle $C$ is left. Observe that if $\Pi_1, \Pi_2, \ldots, \Pi_r$ is the order in which the maximal chains were eliminated to get cycle $C$, then $(C, \Pi_r, \ldots, \Pi_2, \Pi_1)$ denotes the corresponding ear decomposition. For any two-connected graph $G$ and for any cycle $C$, $G$ has an ear decomposition of the form $(C, \Pi_1, \ldots, \Pi_r)$ [13].
Lemma 4.1. Let $G$ be a coneless two-connected simple graph and let $C$ be a cycle of $G$. Then $G$ has an ear decomposition $(C, \Pi_1, \ldots, \Pi_r)$ such that any $\Pi_j$ of length one (if it exists) is a chord of $C$.

Proof. For any ear decomposition $\mathcal{E} = (C, \Pi_1, \ldots, \Pi_r)$ let $p(\mathcal{E})$ denote the number of chains $\Pi_j$ of length one. Assume $\mathcal{E}$ is an ear decomposition with the smallest possible $p(\mathcal{E})$ and suppose there is a chain $\Pi_k = xy$ of length one that is not a chord of $C$. At least one end of $\Pi_k$ is an internal node of some chain $\Pi_i$ with $i < k$, for otherwise $\Pi_k$ would be a chord of $C$.

Case 1: Both $x$ and $y$ are internal nodes of the same $\Pi_i$: let $\Pi_i = v_1, \ldots, v_q$ with $x = v_s, y = v_t$ ($1 < s < t < q$). Replace $\Pi_k$ and $\Pi_i$ by $Q_i = v_1, \ldots, v_s, v_t, \ldots, v_q$ and $R_i = v_s, v_{s+1}, \ldots, v_{t-1}, v_t$ in that order. Observe that $R_i$ is of length at least three since $G$ is simple and coneless. The resulting ear decomposition $\mathcal{E}'$ has $p(\mathcal{E}') < p(\mathcal{E})$, a contradiction.

Case 2: One end of $\Pi_k$, say $x$, is a node on some $\Pi_i$ or $C$ and the other end of $\Pi_k$, namely $y$, is an internal node on some $\Pi_j$ for $j < k$. We choose $i$ as small
as possible, and consequently \( i < j \), or else we are back in Case 1. Let \( \Pi_j = v_1, \ldots, v_q \) and \( y = v_t(1 < t < q) \). Since \( G \) is coneless, we must have either \( t > 2 \) or \( t < q - 1 \). (If \( 2 = t = q - 1 \), then \( q = 3 \) and \( \Pi_j \) has length 2. This is not possible since \( G \) is coneless.) Assume without loss of generality that \( t > 2 \). Replace \( \Pi_j \) by chains \( Q_j = v_q, v_{q-1}, \ldots, v_t = y, x \) and \( R_j = v_1, \ldots, v_t = y \) in this order and delete \( \Pi_k \). The resulting ear decomposition \( \mathcal{E}' \) has \( p(\mathcal{E}') < p(\mathcal{E}) \), which is a contradiction.

**Proposition 4.2.** For a simple bipartite graph \( G \), the following statements are equivalent:

1. \( G \) is coneless,
2. \( G \) has the EC1 property.

**Proof.** (2) \( \Rightarrow \) (1): Clearly if \( G \) contains a cone \( Q \) consisting of a cycle \( C \) with an additional chain \( P \) of length 2, a bicoloring of the edges of \( C \) cannot be extended to an edge 3-coloring of \( Q \). So \( Q \) does not satisfy EC1.

(1) \( \Rightarrow \) (2): We may assume that \( H = G \) and that \( G \) is two-connected and consider an ear decomposition \( \mathcal{E} = (C, \Pi_1, \ldots, \Pi_r) \) of \( G \) such that the only chains \( \Pi_i \) of length one (if any) are chords of \( C \) according to Lemma 4.1. Assume \( \Delta(G) \geq 3 \) (otherwise we are done). The bicoloring of \( C \) can be extended to the rest of \( G \) recursively by starting from \( \Pi_1 \): assume the coloring of \( C \) has been extended to \( \Pi_1, \ldots, \Pi_{k-1} \) and let \( \Pi_k = v_0, \ldots, v_q \). We shall show that the coloring can be extended to \( \Pi_k \) and the result will follow by induction on \( k \).

**Case 1:** \( \Pi_k \) has length at least three:

**Case 1.1:** Both \([v_0, v_1]\) and \([v_{q-1}, v_q]\) can be colored with colors \( c, c' \geq 3 \). Clearly the rest of \( \Pi_k \) can be colored with colors 1 and 2.

**Case 1.2:** One of \([v_0, v_1]\), \([v_{q-1}, v_q]\), say \([v_0, v_1]\), can be colored with 1 or 2 and \([v_{q-1}, v_q]\) can be colored with \( c \geq 3 \). Color \([v_{q-1}, v_q]\) with \( c \) and then starting from \([v_0, v_1]\), color the rest of \( \Pi_k \) with colors 1 and 2.

**Case 1.3:** Each one of \([v_0, v_1]\), \([v_{q-1}, v_q]\) can only be colored with color 1 or 2. Color \([v_{q-1}, v_q]\) accordingly and color \([v_{q-2}, v_{q-1}]\) with color 3. Then starting with \([v_0, v_1]\) color the rest of \( \Pi_k \) with colors 1 and 2.

**Case 2:** \( \Pi_k \) has length one: this is the only remaining case because \( G \) is coneless and hence no \( \Pi_k \) can have length two. If some color \( c \) is missing at both ends \( v_0, v_1 \) of \( \Pi_k \), then color \([v_0, v_1]\) with \( c \). Otherwise, color \( c \) is missing at \( v_0 \) but not color \( d \), and \( d \) is missing at \( v_1 \) but not \( c \). Since \( \Pi_k \) is a chord of \( C \) both \( c \) and \( d \) are different from 1, 2. There is an alternating chain \( P \) with colors \( c, d \) starting from \( v_0 \) in the currently colored graph. We extend it as far as possible. It does not end at \( v_1 \) because \( \Pi_k \cup [v_0, v_1] \) would be an odd cycle, contradicting the fact that \( G \) is bipartite. By interchanging \( c \) and \( d \) along \( \Pi \) we get a coloring where \( d \) is missing at both \( v_0 \) and \( v_1 \). Then color \([v_0, v_1]\) with \( d \) and so \( \Pi_k \) is colored.

We remark that the complete bipartite graph \( K_{2,4} \) is not coneless, yet every edge bicoloring of every cycle can be extended to an edge 4-coloring of the full graph. The
above result is a characterization of the configurations $\mathcal{C}$ of open shop scheduling problems (where one restricts all values $p_{ij}$ to be 0 or 1) for which any preassignment of colors 1, 2 corresponding to a cycle in the associated graph $H$ can be extended to a complete schedule in $\Delta(H)$ time units.

It has led to the class of coneless graphs (these are by definition simple graphs) which strictly contains the class of simple BOC graphs. Notice that BOC graphs may be multigraphs.

5. Concluding remarks

Preassignments occur frequently in timetabling problems as well as in some types of scheduling problems; the special cases studied here are very limited and do not cover by far all the situations occurring in applications. NP-completeness results prevent us from handling efficiently the general case. Many other restricted preassignments may lead to polynomial solutions; further research will undoubtedly shed more light on these cases.

Acknowledgements

The authors would like to thank Dr. D. Coppersmith at IBM T.J. Watson Research Center who pointed out the algorithm based on systems of distinct representatives for handling the case of trees in Section 2. They also express their gratitude to a referee who suggested some improvements in the presentation.

Note added in proof

Regarding restricted edge coloring of trees, we have been made aware that a related problem, restricted node coloring of block graphs, has been addressed by Groeflin in an unpublished report [8], where he gives a characterization of colorability and a polynomial coloring algorithm.

References