Mathematical Methods of Operations Research (1997) 45:145–160

Mixed Graph Colorings

PIERRE HANSEN

Ecole des Hautes Etudes Commerciales, GERAD, Montréal (Québec), Canada

JULIO KUPLINSKY

6, Washington Drive, Ramsey, NJ 07446, USA

DOMINIQUE DE WERRA

Ecole Polytechnique Fédérale de Lausanne, Départment de Mathématiques,
Chaire de Recherche Operationelle, 1015 Lausanne, Switzerland

Abstract: A mixed graph $G_\emptyset$ contains both undirected edges and directed arcs. A k-coloring of $G_\emptyset$ is an assignment to its vertices of integers not exceeding k (also called colors) so that the endvertices of an edge have different colors and the tail of any arc has a smaller color than its head. The chromatic number $\gamma_\emptyset(G)$ of a mixed graph is the smallest k such that $G_\emptyset$ admits a k-coloring. To the best of our knowledge it is studied here for the first time. We present bounds of $\gamma_\emptyset(G)$, discuss algorithms to find this quantity for trees and general graphs, and report computational experience.

Key Words: Graph coloring, oriented graphs, chromatic scheduling.

1 Introduction

Coloring models have often been used as a basic tool for dealing with some special types of scheduling problems which are consequently called chromatic scheduling problems: items (corresponding to the nodes of a graph) have to be scheduled while taking into account some incompatibility requirements (item i cannot be scheduled at the same period as item j) represented by the edges of the graph.

The basic coloring models are too limited for handling the various requirements which are present in real scheduling problems; it is therefore necessary to extend the coloring models in various directions.

We shall in this paper consider a model which can be used for formulating some scheduling problems where both incompatibility requirements and precedence constraints can be present: an arc (i, j) from item i to item j will mean that i must be scheduled before j.

Some fields of application will be mentioned later. Let us first start with some graph theoretical preliminaries.

1432–2994/97/45:1/145–160 $2.50

The idea of orienting some edges of a graph to be colored is rather natural since colorings and circuit-free orientations of graphs are closely related: indeed the nodes of a graph G can be colored with k colors if and only if there is a circuit-free orientation of the edges of G such that no path has more than k nodes. This is the theorem of Roy-Gallai [GAL, ROYb].

We consider finite graphs with no multiple edges or loops. Let $G = (X, E)$ be a graph and $\mathcal{O}$ an orientation of some of its edges (we regard $\mathcal{O}$ as a subset of $\{(i,j) \in X \times X: \{i,j\} \in E\}$). Then $G_{\mathcal{O}} = (X, E, \mathcal{O})$ is called a mixed graph. We will refer to the elements $\{i,j\}$ of E such that neither (i,j) nor (j, i) belong to $\mathcal{O}$ as the unoriented edges of $G_{\mathcal{O}}$.

A k-coloring of $G_{\mathcal{O}}$ is a function $c: X \to \{1, \ldots, k\}$ such that $c(i) \neq c(j)$ for $\{i,j\} \in E$ and $c(i) < c(j)$ for $(i,j) \in \mathcal{O}$. Clearly such a coloring may not always exist: if $\mathcal{O}$ contains a circuit, there is no k-coloring for any k.

This extension of the concept of vertex coloring provides a model for some types of exam scheduling, e.g. when written exams have to be taken before oral ones, in addition to the usual constraint that no student can take two or more exams during the same period. More generally, some types of chromatic scheduling problems can be handled with this model. Let T be a collection of jobs (with unit processing times). These jobs have to be processed taking into account the following constraints.

1. *Precedence constraints.* There is a set of ordered pairs of jobs (i,j) such that i must be processed before j.
2. *Disjunctive constraints.* For some collection $\{I_\alpha: \alpha \in A\}$ of subsets of X, no two jobs in I_α can be processed simultaneously.

Consider now a mixed graph $G_{\mathcal{O}} = (X, E, \mathcal{O})$ obtained as follows:

1. To each job j in T we associate a vertex j in X. G has no other vertices, no arcs and no edges.
2. For each ordered pair (i,j) of jobs we introduce an arc (i,j) in $G_{\mathcal{O}}$.
3. For each subset I_α we introduce a clique associated with the jobs in I_α. (If an edge is needed between vertices i and j, we introduce it only if there was no previous arc or edge joining i and j).

Now there is a one-to-one correspondence between feasible schedules in k time units and k-colorings of the mixed graph $G_{\mathcal{O}}$.

Notice that scheduling problems with disjunctive constraints usually have some additional structure, e.g. the directed subgraph may consist of disjoint paths from a source to a sink. Branch and bound and 0-1 programming algorithms for minimum makespan in such graphs (e.g. [ROYa], [BAL]) could be adapted to the mixed graph coloring problem but might be less efficient than specially tailored ones.

2 Bounds on the Chromatic Number

We first study when a mixed graph has a coloring.

Definitions: If G is a graph, an orientation $\mathcal{O}$ of its edges is called *complete* iff for every edge $\{i, j\}$ of G either $(i, j) \in \mathcal{O}$ or $(j, i) \in \mathcal{O}$.

For a vertex i of a mixed graph, the length $r(i)$ of a longest directed path ending at i is called the *inrank* of i, while the length of a longest directed path starting at i is the *outrank* of i.

Proposition 1:

(i) A mixed graph and $G_{\mathcal{O}}$ has a k-coloring iff there is a complete circuit-free orientation $\mathcal{O}' \supseteq \mathcal{O}$ in E such that all directed paths have length at most $k - 1$.
(ii) $G_{\mathcal{O}}$ has a coloring iff $\mathcal{O}$ contains no circuits.

Proof of Proposition 1:

(i) Suppose that $G_{\mathcal{O}}$ has a k-coloring. For each unoriented edge choose the orientation from the lower color to the upper color. Obviously this orientation has no circuits. The vertices of a directed path will have increasing colors, so the length of a path will be at most $k - 1$. Conversely, suppose such an orientation exists. If (i, j) is an arc, we clearly have $r(i) < r(j)$, so $\{r(i) + 1\}_{i=1}^{|X|}$ is a k-coloring of $G_{\mathcal{O}}$.
(ii) If $\mathcal{O}$ contains no circuit, we can find a circuit-free orientation $\mathcal{O}' \supseteq \mathcal{O}$ which is complete: define a numbering $n_1, \ldots, n_{|X|}$ of the vertices such that $n_i < n_j$ if $(i, j) \in \mathcal{O}$. Orient every unoriented edge $\{i, j\}$ from i to j if $n_i < n_j$ and from j to i otherwise. This defines a circuit-free orientation $\mathcal{O}'$ of the required type. Then from (i) a node coloring exists for $G_{\mathcal{O}}$. The rest follows also from (i).

The idea of defining a coloring on the basis of the inranks is due to Vitaver [VIT].

Definitions:

1. Given a graph G, the set of circuit-free orientations of G will be denoted $\Omega(G)$ or simply Ω.
2. Given a graph G and $\mathcal{O} \in \Omega$, the smallest k such that there exists a k-coloring of G will be called the chromatic number of $G_{\mathcal{O}}$ and will be denoted by $\gamma_{\mathcal{O}}(G)$.

Notice that $\gamma_{\varnothing}(G)$ is the usual chromatic number of the graph G; it is denoted by $\gamma(G)$.

Using the concept of chromatic number, we can reformulate part of Proposition 1 as an extension of the classical Roy-Gallai theorem [ROYb, GAL] .

Proposition 2: Let G be a graph and $\mathcal{O} \in \Omega$. We denote by $\Omega_{\mathcal{O}}$ the set of complete circuit-free orientations of G containing $\mathcal{O}$. Then

$$\gamma_{\mathcal{O}}(G) = 1 + \min_{\substack{\mathcal{O} \subset \mathcal{O}' \\ \mathcal{O}' \in \Omega}} \ell(\mathcal{O}') \ ,$$

where $\ell(\mathcal{O}')$ is the largest inrank of $G_{\mathcal{O}'}$.

Proposition 3: For a connected graph G, $\gamma_{\mathcal{O}}(G) = \gamma(G)$ for every $\mathcal{O} \in \Omega$ iff G is complete.

Proof: Suppose that G is complete. For every $\mathcal{O} \in \Omega$ we clearly have $\gamma_{\mathcal{O}}(G) \geq \gamma(G) = n$, where $n = |X(G)|$. Now, $\mathcal{O}$ can be extended to a complete orientation $\mathcal{O}' \in \Omega$, so we obtain $n = \gamma_{\mathcal{O}'}(G) \geq \gamma_{\mathcal{O}}(G) \geq \gamma(G) = n$.

Suppose now that $\gamma_{\mathcal{O}}(G) = \gamma(G) = k > 1$ for all $\mathcal{O}$ and consider a k-coloring of G. We can assume that this coloring has been obtained by a sequential coloring procedure (i.e. vertices have been ordered and assigned successively the first color not given to an adjacent vertex in such a way that k colors are used, see for instance [MMI]). So take a vertex x_k having color k. It is adjacent to a vertex x_{k-1} having color $k - 1$. This is in turn adjacent to a vertex x_{k-2} having color $k - 2$. Continuing in this way we obtain a path $x_1, \ldots, x_k$.

If G only contained the vertices $(x_1, \ldots, x_k)$, then $k = n$ and G would be a complete graph (for otherwise we would have $\gamma(G) < n$). So assume that G contains a vertex $x \notin \{x_1, \ldots, x_k\}$ adjacent to some vertex x_i.

Notice that x_k must be adjacent to a vertex with color 1. This vertex must be x_1, for otherwise we would have a path of length k in G, and we could find a circuit-free orientation containing a directed path of length k, contradicting Proposition 1. But then $(x, x_i, \ldots, x_k, x_1, \ldots, x_{i-1})$ is a path of length k, and we reach again a contradiction.

Notation: Given a graph G and $\mathcal{O} \in \Omega$, we denote by $X_{\mathcal{O}}$, the set of vertices of $G_{\mathcal{O}}$ which are incident to some arc in $\mathcal{O}$, and by $\overline{G}_{\mathcal{O}}$ the subgraph of G generated by the vertices in $X_{\mathcal{O}}$ (the arcs in $\mathcal{O}$ are considered as edges in $\overline{G}_{\mathcal{O}}$).

Proposition 4: Let G be a graph and $\mathcal{O} \in \Omega$. Then

$$\gamma_{\mathcal{O}}(G) \leq \gamma(G) + |X_{\mathcal{O}}| - \gamma(\overline{G}_{\mathcal{O}}).$$

Proof: Let $k = \gamma(G)$ and take a k-coloring c of G. We will now modify c so as to obtain a coloring of $G_{\mathcal{O}}$. Let $Y_1 = \{i \notin X_{\mathcal{O}}: c(i) = c(i') \text{ for some } i' \in X_{\mathcal{O}}\}$, and let Y_2 be the set of vertices not in $X_{\mathcal{O}}$ or Y_1. We let x denote the number of vertices in $X_{\mathcal{O}}$.

The new coloring d is defined in the following way. We order the vertices in $X_{\mathcal{O}}$ according to non decreasing inranks introducing a new color for each vertex. Now for every $i \in Y_1$ we choose some $i' \in X_{\mathcal{O}}$ so that $c(i) = c(i')$ and then define $d(i) = d(i')$. Notice that, since $c(X_{\mathcal{O}} \cup Y_1 \cup Y_2) = c(X_{\mathcal{O}}) \cup c(Y_2)$ and $c(X_{\mathcal{O}})$ and $c(Y_2)$ have no common elements, we have

$$|c(X_{\mathcal{O}})| + |c(Y_2)| = k\ . \tag{1}$$

Finally the colors in the set $c(Y_2)$ are changed to the colors $\{x + 1, \ldots, x + h\}$, where $h = |c(Y_2)|$.

We claim that d is a coloring of $G_{\mathcal{O}}$. In fact, if (i, j) is an arc, then $i, j \in X_{\mathcal{O}}$, and the inrank of i is less than the inrank of j, so $d(i) < d(j)$. Suppose now that $\{i, j\}$ is an edge. Since the colors of the vertices in $X_{\mathcal{O}}$ are all different, we may assume that $i, j \in Y_1 \cup Y_2$ or $i \in Y_1 \cup Y_2$ and $j \in X_{\mathcal{O}}$. Now the vertices in $X_{\mathcal{O}}$ and Y_1 are colored with the elements in $\{1, \ldots, x\}$, while those in Y_2 are colored with $\{x + 1, \ldots, x + h\}$, and moreover $d(Y_2)$ is a recoloring of $c(Y_2)$, so we need only consider the cases $i, j \in Y_1$ and $i \in Y_1$, $j \in X_{\mathcal{O}}$. In both cases $c(i) = c(i')$ and $c(j) = c(j')$ for some $i', j' \in X_{\mathcal{O}}$, and $d(i) = d(i')$, $d(j) = d(j')$. Suppose $d(i) = d(j)$, then $i' = j'$, so $c(i) = c(j)$, a contradiction.

We now count the number of colors used by d. By virtue of equation (1) we have $|d(X_{\mathcal{O}} \cup Y_1 \cup Y_2)| = x + h = x + k - |c(X_{\mathcal{O}})|$. But $c|_{X_{\mathcal{O}}}$ is a coloring of $\overline{G}_{\mathcal{O}}$, so $|c(X_{\mathcal{O}})| \geq \gamma(\overline{G}_{\mathcal{O}})$, and the number of colors used by d is at most $|X_{\mathcal{O}}| + \gamma(G) - \gamma(\overline{G}_{\mathcal{O}})$.

Remarks:

i. Using this result, we conclude again that if $G_{\mathcal{O}}$ is a clique (complete graph), then $\gamma_{\mathcal{O}}(G) = \gamma(G)$. A more general case in which $\gamma_{\mathcal{O}}(G) = \gamma(G)$ occurs when G is a comparability graph (transitively orientable graph) and $\mathcal{O}$ is a partial orientation which can be extended to a transitive orientation. A trivial case in which $\gamma_{\mathcal{O}}(G) = \gamma(G)$ is that of a bipartite graph $G = (X, Y, E)$ when $\mathcal{O}$ consists of an arbitrary collection of arcs oriented from X to Y.
ii. The bound of Proposition 4 is best possible in the following sense. For any p there exists a graph G and an orientation $\mathcal{O} \in \Omega$ with $|X_{\mathcal{O}}| - \gamma(\overline{G}_{\mathcal{O}}) = p$ sat-

isfying $\gamma_{\mathcal{O}}(G) = \gamma(G) + p$. Take for instance for G a chain of $p + 1$ edges and choose $\mathcal{O}$ so that $G_{\mathcal{O}}$ is a directed path consisting of $p + 1$ acrs. Then $|X_{\mathcal{O}}| = \gamma_{\mathcal{O}}(G) = p + 2$ and $\gamma(G) = \gamma(\overline{G}_{\mathcal{O}}) = 2$.

Proposition 5: For any graph G and any $\mathcal{O} \in \Omega$, $\mathcal{O} \neq \emptyset$,

$$\gamma_{\mathcal{O}}(G) \leq \gamma_{\mathcal{O}}(G^*)(\gamma(G) - 1) + 1 \ .$$

Here G^* denotes the directed partial subgraph $(X_{\mathcal{O}}, \mathcal{O})$ of $G_{\mathcal{O}}$.

Proof: Let $k = \gamma(G)$ and consider a coloring c of G using k colors. We will modify c so as to obtain a coloring d of $G_{\mathcal{O}}$. For $i \in G^*$ let $r(i)$ be the inrank of i in G^*, so that $\gamma_{\mathcal{O}}(G^*) = \max\{r(i): i \in G^*\} + 1$. We color the vertices of G^* according to non decreasing inranks. The vertices of inrank 0 keep the color they have by c. Let us assume now that all vertices with inrank less than s have been assigned a new color and let $r(i) = s$. If $c(i) = 1$, then $d(i) = sk + 1$, if $c(i) = 2$, then $d(i) = sk + 2$. We proceed in this way, with one exception. Suppose that the highest value of d for a vertex of inrank $s - 1$ occurs for a vertex j and $c(j) = h$. Then if $c(i) = h$, we define $d(i) = d(j) \leq (s - 1)k + h$. In this way we use at most k colors for vertices of inrank 0 and at most $k - 1$ new colors for every other inrank. For vertices not in G^* we let d coincide with c.

Notice that if $i, j \in G^*$ and $r(j) = r(i) + 1$, then $d(i) \leq d(j)$, and in fact $d(i) < d(j)$ unless $c(i) = c(j)$. We conclude from this remark that if (i, j) is an arc in $\mathcal{O}$ (so that $r(i) < r(j)$), then $d(i) < d(j)$, since $\{i, j\} \in E$. Let now $\{i, j\}$ be an edge. We conclude similarly that $d(i) \neq d(j)$ if $i, j \in G^*$, and obviously the same holds if $i, j \notin G^*$, so we may assume that $i \in G^*, j \notin G^*$. If $d(i) \leq k$, then $d(i) = c(i) \neq c(j) = d(j)$, so in any case $d(i) \neq d(j)$.

Notice that d uses at most k colors on vertices of inrank 0, and that, for $s > 0$, the set $\{d(i): r(i) = s\} \setminus \{d(i): r(i) < s\}$ has at most $k - 1$ elements. Since $\gamma_{\mathcal{O}}(G^*) \geq 1$, we conclude that d uses at most $\gamma(G) + (\gamma_{\mathcal{O}}(G^*) - 1)(\gamma(G) - 1) = \gamma_{\mathcal{O}}(G^*)(\gamma(G) - 1) + 1$ colors.

Remark: The bound of Proposition 5 is best possible. Take for example r copies $K^1, \ldots, K^r$ of a clique with s vertices. Let x_{ij} $(i = 1, \ldots, s)$ be the vertices of K^j and for $u = 1, \ldots, r - 1$ introduce arcs $(x_{i,u}, x_{j,u+1})$ where $i \neq j$. The mixed graph $G_{\mathcal{O}}$ thus obtained satisfies

$$\gamma_{\mathcal{O}}(G) = s + (s - 1)(r - 1) = r(s - 1) + 1 \ .$$

Notice that $\gamma_{\mathcal{O}}(G^*) = r$. Since the chromatic number of the underlying graph is s (vertices x_{ij} may get color i for $j = 1, \ldots, r$), we have

$$\gamma_{\mathcal{O}}(G) = \gamma_{\mathcal{O}}(G^*)(\gamma(G) - 1) + 1 \ .$$

3 Coloring Algorithms

Proposition 2 suggests a possible coloring algorithm for mixed graphs. One need only devise a branch-and-bound procedure in which the branchings represent the two possible orientations of an unoriented edge, and the bounds are given by the lengths of the longest directed paths. Our computational experience with such an algorithm has not been promising though, so we will present later in this paper a more direct approach.

As a first approach to coloring a mixed graph, we classify unoriented edges e locally. Let e be an unoriented edge such that neither of its orientations creates a circuit. We say that e is a 0-(1-, 2-) edge iff 0 (resp. 1, 2) of its possible orientations increase the length of the longest directed path of $G_{\mathcal{O}}$. We present this concept more formally. Here $G^* \cup e$ denotes the graph G^* to which edge e has been added.

Definition: Let a, $-a$ be the two possible orientations of e, $\mathcal{O}^+ = \mathcal{O} \cup \{a\}$, $\mathcal{O}^- = \mathcal{O} \cup \{-a\}$. Then

e is a 0-edge iff $\gamma_{\mathcal{O}^+}(G^* \cup e) = \gamma_{\mathcal{O}^-}(G^* \cup e) = \gamma_{\mathcal{O}}(G^*)$,
e is a 1-edge iff $\gamma_{\mathcal{O}^+}(G^* \cup e) > \gamma_{\mathcal{O}}(G^*)$ or $\gamma_{\mathcal{O}^-}(G^* \cup e)$, $> \gamma_{\mathcal{O}}(G^*)$, but not both, and
e is a 2-edge iff $\gamma_{\mathcal{O}^+}(G^* \cup e) > \gamma_{\mathcal{O}}(G^*)$ and $\gamma_{\mathcal{O}^-}(G^* \cup e) > \gamma_{\mathcal{O}}(G^*)$.

As in Proposition 5, G^* is the directed graph consisting of the arcs in $\mathcal{O}$ and the vertices incident to them.

Remarks 6:

i) A 0-edge may not be assigned an arbitrary orientation, i.e. it may happen that $\gamma_{\mathcal{O}^+}(G) > \gamma_{\mathcal{O}}(G)$. Consider for instance the mixed graph in Figure 1. Orienting e upwards does not change $\gamma_{\mathcal{O}}(G^*) = 3$, but increases $\gamma_{\mathcal{O}}(G)$ to 4.

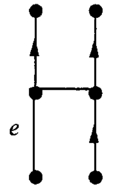

Fig. 1. A 0-edge

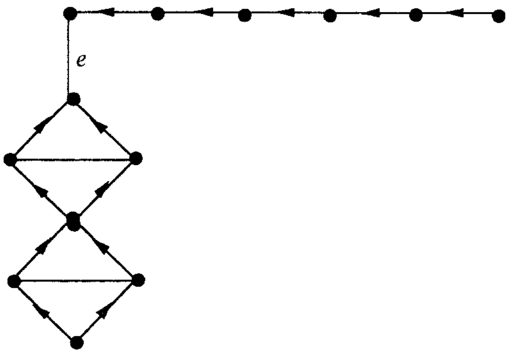

Fig. 2. A 1-edge

ii) Similarly, assigning to a 1-edge the orientation that does not increase the maximum inrank may be incompatible with an optimal coloring. This is shown by the mixed graph in Figure 2. Clearly, e is a 1-edge. A greedy algorithm would orient e upwards. But then completing the orientation arbitrarily one obtains a directed path of length 7. However, reversing the orientation of e in this orientation one obtains a complete orientation in which the maximum inrank is 6.

iii) Eliminating the rightmost vertex in Figure 2 we obtain a similar example in which e is a 2-edge.

Proposition 7: Let e be a 2-edge in $G_{\mathcal{O}}$.

a) Both orientations of e increase the maximum inrank by exactly 1.
b) The endvertices of e have the same inrank and the same outrank.

Proof:

a) Let ℓ be the maximum inrank of $G_{\mathcal{O}}$, and let a, $-a$ be the two orientations of e. We write $\mathcal{O}^+ = \mathcal{O} \cup \{a\}$ and $\mathcal{O}^- = \mathcal{O} \cup \{-a\}$. Since e is a 2-edge, the maximum inrank of $G_{\mathcal{O}^+}$ is greater than ℓ; it is attained by a directed path (σ, a, σ'), where σ and σ' are directed paths in $G_{\mathcal{O}}$. Hence $L(\sigma) + L(\sigma') \geq \ell$, where $L(\cdot)$ denotes the length of a directed path. If a has tail i and head j, we clearly have that $L(\sigma)$ is the inrank of i and $L(\sigma')$ is the outrank of j in $G_{\mathcal{O}}$. Similarly, the maximum inrank of $G_{\mathcal{O}^-}$ is attained by a directed path $(\tau, -a, \tau')$ where τ and τ' are directed paths in $G_{\mathcal{O}}$ with $L(\tau) + L(\tau') \geq \ell$. Now $L(\tau)$ is the inrank of j and $L(\tau')$ is the outrank of i. Adding the inequalities we obtain $2\ell \leq L(\sigma) + L(\sigma') + L(\tau) + L(\tau') \leq 2\ell$, since (τ, σ') is a directed path of length $L(\tau) + L(\sigma') \leq \ell$, and similarly for (σ, τ'). Hence we must have $L(\sigma) + L(\sigma') = L(\tau) + L(\tau') = \ell$. This proves that both orientations of e increase the maximum inrank by exactly 1.

b) Adding again $L(\tau) + L(\sigma') \leq \ell$ and $L(\sigma) + L(\tau') \leq \ell$ we obtain $2\ell = L(\sigma) + L(\sigma') + L(\tau) + L(\tau') \leq 2\ell$, and hence $L(\tau) + L(\sigma') = L(\sigma) + L(\tau') = \ell$, so $L(\sigma) = L(\tau)$ and $L(\sigma') = L(\tau')$.

Case of trees

Some results in this subsection will hold for arbitrary bipartite graphs, but we will focus primarily on trees.

By Proposition 5, if G is bipartite and $\mathcal{O} \in \Omega$, then $\gamma_{\mathcal{O}}(G) \leq \gamma_{\mathcal{O}}(G^*) + 1$, so we have only two possibilities for the chromatic number of $G_{\mathcal{O}}$.

Definition: Let G be a bipartite graph and $\mathcal{O} \in \Omega$. We say that $G_{\mathcal{O}}$ is *short* iff $\gamma_{\mathcal{O}}(G) = \gamma_{\mathcal{O}}(G^*)$, and *long* iff $\gamma_{\mathcal{O}}(G) = \gamma_{\mathcal{O}}(G^*) + 1$.

A long mixed graph is the one of Fig. 1 with edge e oriented upwards.

0-2 Algorithm: Let G be a tree and $\mathcal{O} \in \Omega$ such that $G_{\mathcal{O}}$ only has 0- or 2-edges. Then the following algorithm provides an optimal orientation of $G_{\mathcal{O}}$, i.e., an orientation giving an optimal coloring.

Let G be rooted at an arbitrary vertex r. A Boolean variable $f \in \{\text{up}, \text{down}\}$, a parameter, is set originally equal to up. We traverse the tree in depth-first order (preorder, see [TAR]). Each time we encounter an unoriented edge we orient it according to f if it hasn't been assigned an orientation before and switch f in any case.

If e is an unoriented edge of $G_{\mathcal{O}}$ then we will refer to the path from r to one of the vertices of e that does not contain the other as the path from r to e. The orientation that the algorithm assigns to e is determined by the number of unoriented edges contained in the path from r to e: the edge is oriented up if this number is even and down if it is odd.

The main property of the orientation $\mathcal{P}$ given by the algorithm is that no directed path of $\mathcal{P}$ contains more than one unoriented edge. In fact, let e', e'' belong to a directed path of $\mathcal{P}$, and assume without loss of generality that all arcs in the path which are between e' and e'' belong to $\mathcal{O}$. This directed path will be denoted by $(x_1, \ldots, x_s)$ and we will assume that $e' = \{x_1, x_2\}$, $e'' = \{x_{s-1}, x_s\}$. For brevity we will set $\overline{X} = \{x_1, \ldots, x_s\}$. Let $(y_1, \ldots, y_p)$ be the path from r to e', and let i be minimum such that $y_i \in \overline{X}$, say $y_i = x_t$. Notice that $y_{i+1}, \ldots, y_p \in \overline{X}$, for otherwise $(x_t, \ldots, x_2)$ and $(y_i, \ldots, y_p)$ would be two different paths from x_t to x_2.

Assume first that $s > t > 1$. In this case the number of unoriented edges in the path from r to e' is equal to the number of unoriented edges in $(y_1, \ldots, y_i)$ and must be even. On the other hand, the path from r to e'' is $(y_1, \ldots, y_i, x_{t+1}, \ldots, x_{s-1})$, and the number of unoriented edges in this path is equal to the number of unoriented edges in $(y_1, \ldots, y_i)$, and must be odd, a contradiction.

If $t = 1$, a similar argument shows that the orientation of e' forces the number of unoriented edges in $(y_1, \ldots, y_p)$ to be odd. But now the path between r and e'' is $(y_1, \ldots, y_p, x_2, \ldots, x_{s-1})$, which contains one more unoriented edge than $(y_1, \ldots, y_p)$, and again we reach a contradiction.

If $t = s$, again a similar argument shows that the orientation of e'' forces the number of unoriented edges in $\{y_1, \ldots, y_p\}$ to be even. But then the path between r and e' is $\{y_1, \ldots, y_p, x_s, \ldots, x_2\}$ which contains one more unoriented edge than $\{y_1, \ldots, y_p\}$ and once again we reach a contradiction.

Let ℓ be the length of the longest directed path in G^*. Then, if $G_{\mathcal{O}}$ has only 0-edges, $\mathcal{P}$ cannot contain a directed path of length $\ell + 1$, and if it also has 2-edges, by Proposition 7, it cannot contain a directed path of length $\ell + 2$.

If G is an acyclic digraph having m edges, the inranks of its vertices can be obtained in $O(m)$ time. It follows that the 0-2 algorithm has $O(n)$ complexity.

Corollary 8: Let G be a tree and $\mathcal{O} \in \Omega$. If $G_{\mathcal{O}}$ has only 0-edges, then it is short.

Remark: Replacing "0" by "0 or 1" in the above corollary does not result in a generally true statement. Consider for instance the graph in Figure 3 where every unoriented edge is a 1-edge.

The following result, which we need for our algorithm, sheds some light on Remark 6 ii).

Proposition 9: Let G be bipartite and $\mathcal{O} \in \Omega$. Suppose that e is a 1-edge and a is the orientation that does not increase the maximum inrank in $G_{\mathcal{O}}$. If we denote $\mathcal{O}^+ = \mathcal{O} \cup \{a\}$, then $\gamma_{\mathcal{O}^+}(G) = \gamma_{\mathcal{O}}(G)$.

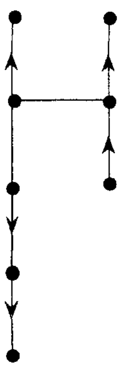

Fig. 3. A tree with 1-edges

Proof: If $G_{\mathcal{O}}$ is short, then $\gamma_{\mathcal{O}}(G) = \gamma_{\mathcal{O}}(G^*)$. If $\mathcal{P}$ is a complete circuit-free orientation containing $\mathcal{O}$ which realizes the chromatic number, then $a \in \mathcal{P}$, so $\gamma_{\mathcal{O}^+}(G) \leq \gamma_{\mathcal{O}}(G)$.

If $G_{\mathcal{O}}$ is long, since $\gamma_{\mathcal{O}} + (G^*) = \gamma_{\mathcal{O}}(G^*)$, we have by Proposition 5 $\gamma_{\mathcal{O}} + (G) \leq \gamma_{\mathcal{O}}(G^*) + 1 = \gamma_{\mathcal{O}}(G)$.

Definition: Let e be a 1-edge in $G_{\mathcal{O}}$. The orientation of e that does not increase the maximum inrank of $G_{\mathcal{O}}$ is called the *conformal* orientation of e.

Tree Algorithm: If G is a tree with more than one vertex and $\mathcal{O} \in \Omega$ the following algorithm provides an optimal coloring of $G_{\mathcal{O}}$.

1. If $\mathcal{O} = \varnothing$ traverse the tree coloring alternately the vertices with two colors and exit. Otherwise go to 2.
2. Classify all unoriented edges as 0-, 1-, or 2-edges.
3. If the graph contains a 2-edge, go to 6. Otherwise go to 4.
4. If there are no 1-edges go to 7. Otherwise go to 5.
5. Orient one arbitrary 1-edge conformally. Go to 2.
6. Apply the method of Proposition 5 to obtain a coloring with $\gamma_{\mathcal{O}}(G^*) + 1$ colors and exit.
7. Apply the 0-2 algorithm and exit.

Proposition 10: The previous algorithm results in an optimal coloring of $G_{\mathcal{O}}$ when G is a nontrivial tree.

Proof: Note that as soon as we know that $G_{\mathcal{O}}$ is long, applying the construction of Proposition 5 will result in an optimal coloring. This justifies the two instances of the algorithm where step 6 is invoked.

Since the 0-2 algorithm has complexity $O(n)$, this algorithm has complexity $O(n^2)$. We remark that the complexity depends basically on the number of 1-edges that may appear during the execution of the algorithm. The next two results which are valid for general mixed graphs, give some information about this number.

Proposition 11: Let $(e_1, \ldots, e_k)$ be a path in $G_{\mathcal{O}}$, where $e_i = \{x_{i-1}, x_i\}$ is a 1-edge for each i. Suppose that each orientation (x_{i-1}, x_i) increases the maximum inrank. Then $k \leq 2\ell + 1$, where ℓ is the maximum inrank of $G_{\mathcal{O}}$.

Proof: By assumption, for $1 \leq i \leq k$ there exist directed paths σ_i ending at x_{i-1} and τ_i starting at x_i such that, in the notation of Proposition 8, $L(\sigma_i) + L(\tau_i) \geq \ell$. We write for brevity $a_i = L(\sigma_i)$, $b_i = L(\tau_i)$, so that the last inequality reads $a_i + b_i \geq \ell$ for $1 \leq i \leq k$. On the other hand, since e_i is a 1-edge, we necessarily have $a_i + b_{i-2} \leq \ell - 1$ for $2 \leq i \leq k$, where we define $b_0 = 0$. It follows that $b_i \geq \ell - a_i \geq b_{i-2} + 1$ for $2 \leq i \leq k$, while $b_0, b_1 \geq 0$. By induction, $i \leq b_{2i} \leq \ell$ if $0 \leq 2i \leq k$, and hence $k \leq 2\ell + 1$.

Proposition 12: Let $G_{\mathcal{O}}$ have 1-edges $e_1, \ldots, e_k$ and suppose that $a_1, \ldots, a_k$ are the conformal orientations of these edges. We set $\mathcal{P} = \mathcal{O} \cup \{a_1, \ldots, a_k\}$. Further, we denote by ℓ the length of a longest directed path of $G_{\mathcal{O}}$ and by L_r the length of a longest directed path of $G_{\mathcal{P}}$ containing r of the arcs $a_1, \ldots, a_k$, where $0 \leq r \leq k$. If $r, s \geq 0$ and $r + s < k$, then

$$L_{r+s+1} \leq L_r + L_s - \ell + 1 \ .$$

Proof: We denote $A = (a_1, \ldots, a_k)$. Let α be a directed path in $G_{\mathcal{P}}$, longest among those containing $r + s + 1$ of the arcs in A. Then α is a concatenation (β, a, γ) where β and γ are directed paths in $G_{\mathcal{P}}$ containing respectively r and s arcs in A, and $a = (x, y) \in A$. On the other hand, since a is the conformal orientation of a 1-edge, there exist oriented paths in $G_{\mathcal{O}}$ σ ending at y and τ starting at x such that

$$L(\sigma) + L(\tau) \geq \ell$$

Note that

$$L(\beta) + L(\tau) \leq L_r$$

and

$$L(\sigma) + L(\gamma) \leq L_s \ .$$

Using the last three inequalities we obtain $L(\beta) + L(\gamma) \leq L_r + L_s - L(\sigma) - L(\tau) \leq L_r + L_s - \ell$.

Corollary 13: Let e_1, e_2, e_3 be 1-edges in a bipartite mixed graph $G_{\mathcal{O}}$. Assume they are the only edges in $G_{\mathcal{O}}$. Then, orienting these edges conformally does not increase the chromatic number of the mixed graph.

Proof: This is certainly true if $G_{\mathcal{O}}$ is short. If it is long, a longest directed path in $G_{\mathcal{P}}$ has length at most $L_3 \leq 2L_1 - \ell + 1$. The result follows from the observation that $L_1 = \ell$.

Remark: The mixed graph in Fig. 4 shows that the last result does not hold for four edges.

Algorithms for general graphs and computational experience

We have used a branch-and-bound algorithm to color a collection of randomly generated graphs. To generate a graph with m_a arcs and m_e edges we choose uniformly at random without repetition $m_a + m_e$ integers $\tau_i \in \{1, \ldots, M\}$, where $M = \binom{n}{2}$. This defines a labeled graph on n vertices having arcs r_i, $1 \leq i \leq m_a$ and edges r_i, $m_a < i \leq m_a + m_e$.

In every stage during the coloring process we have a partial coloring of $G_{\mathcal{O}}$. We use now the fact that assigning a color to a vertex x implies that certain colors will not be available for vertices on directed paths passing through x. We will say that those colors are forbidden for the corresponding vertices. A similar remark applies to neighbors of x, i.e. vertices that are incident to an edge of which x is an endvertex. The next vertex to be colored is one with the least number of forbidden colors. The branching is done according to the color assigned to the current vertex.

Some of the procedures deserve comment. As we mentioned above, we need to keep current the inrank and outrank of x. To exemplify, we describe the calculation of the inranks. Initially, we assign to every vertex x its indegree (the number of arcs which have x as their head). During stage k, we identify the vertices of inrank k (these are those vertices for which the assigned value is 0) and reduce the value assigned to their outneighbors by 1. At the beginning

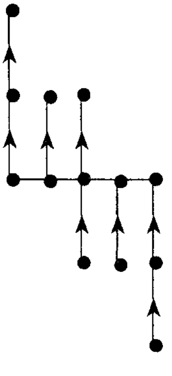

Fig. 4. Four 1-edges which should not be oriented simultaneously

of stage k, the value assigned to the vertices of inrank at least k is the number of inneighbors of inrank at least k. This invariant shows the correctness of the algorithm.

We also have to maintain maxima and minima of the sets of feasible colors for all vertices. To this effect we use min-max heaps, a data structure introduced by Atkinson et al. [ASST]. This structure is similar to a heap, but it has two types of levels: values stored on even (odd) levels are smaller (greater) than or equal to values stored at their descendants. A maximum or a minimum value can be retrieved, and the heap updated, in $O(\log n)$ time, where n is the number of elements in the heap. Creating the heap requires $O(n)$ computations.

The following tables (1 and 2) show sample times needed to color mixed graphs. The graphs in the first table have order 50, and those of the second table have order 70. In both cases we have colored 10 graphs of each edge density/arc density pair. The numbers given are seconds of CPU time per graph on a Sun Station. They are sometimes large, as graph coloring is NP-hard, notoriously difficult in practice; furthermore we use only simple bounds in the algorithm. Notice that adding a few arcs makes the problem easier.

Table 1. Graphs of order 50

Arc densities	Edge densities	
	0.10	0.20
0.0	10	34
0.1	8	13
0.2	11	27
0.3	17	54
0.4	22	89
0.5	20	1866
0.6	58	585
0.7	175	
0.8	523	

Table 2. Graphs of order 70

Arc densities	Edge densities	
	0.10	0.20
0.0	12	6374
0.1	15	47
0.2	19	363
0.3	27	1980
0.4	119	
0.5	629	
0.6	2079	

We observe that there is a large amount of time spent on graphs of order 50 having edge density 0.20 and arc density 0.5; it may have happened that we came across one pathological case which took much time. We have however no explanation why this case should be more difficult than the neighbour cases.

Similarly for graphs of order 70 having no arcs and edge density 0.20 we have no explanation for this pathological behaviour.

Concluding Remarks

Although the model described in this paper seems rather natural for some scheduling problems involving disjunctive as well as sequential constraints, we do not know of a similar approach in the scheduling or graph coloring literatures.

Some questions remain open; among those the complexity of deciding whether for a bipartite mixed graph G, the chromatic number is $\gamma_{\mathcal{O}}(G^*)$ or $\gamma_{\mathcal{O}}(G^*) + 1$.

Besides this more experiments should be carried out on other types of graphs and as far as applications are concerned it would be worth developing heuristic procedures for solving large problems with guarantees of performance. Propositions 4 and 5 are a first step in this direction.

Acknowledgments: This paper was written partially when the third author visited RUTCOR at Rutgers University, New Brunswick, N.J. and partially when the first author visited the Dept of Maths at Ecole Polytechnique Fédérale de Lausanne in 1994–1995. The support of both institutions is acknowledged. The first author was also supported by NSERC grant GP 0105574 and FCAR grant 90 NC 0305. We thank three anonymous referees for useful comments and suggestions.

References

[ASST] Atkinson MD, Sack J-R, Santoro N, Strothotte T (1986) Min-max heaps and generalized priority queues. Communications of the Association for Computing Machinery 29:996–1000

[BAL] Balas E (1969) Machine sequencing via disjunctive graphs: An implicit enumeration approach. Operations Research 17:941–957

[GAL] Gallai T (1968) On directed paths and circuits. In: Erdös P, Katona G (Eds.) Theory of graphs – proceedings of the colloquium held at Tihany, Hungary. Academic Press

[MMI] Matula DW, Marble G, Isaacson JD (1972) Graph coloring algorithms. In: Read RC (Ed.) Graph Theory and Computing. Academic Press

[ROYa] Roy B (1966) Prise en compte des contraintes disjonctives dans les méthodes de chemin critique. RAIRO 38:69–84
[ROYb] Roy B (1967) Nombre chromatique et plus longs chemins d'un graphe. Revue Française d'Informatique et de Recherche Opérationnelle 1:129–132
[TAR] Tarjan RE (1983) Data structures and network algorithms. SIAM
[VIT] Vitaver LM (1962) Determination of minimal colorings for vertices of a graph by means of Boolean powers of the adjacency martix. Soviet Mathematics – Doklady 3, no 6: 1687–1688

Received: April 1995
Revised version received: April 1996