Scheduling problems and mixed graph colorings[1]

Yuri N. Sotskov, Vjacheslav S. Tanaev
Institute of Engineering Cybernetics, Surganov St. 6,
220012 Minsk, Belarus

Frank Werner
Otto-von-Guericke-Universität, Fakultät für Mathematik, PSF 4120,
39016 Magdeburg, Germany

Abstract: Some scheduling problems induce a mixed graph coloring, i.e. an assignment of positive integers (colors) to vertices of a mixed graph such that, if two vertices are joined by an edge, then their colors have to be different, and if two vertices are joined by an arc, then the color of the startvertex has to be not greater than the color of the endvertex. We discuss some algorithms for coloring the vertices of a mixed graph with a small number t of colors and present computational results for calculating the chromatic number, i.e. the minimal possible value of such a t. We also study the chromatic polynomial of a mixed graph which may be used for calculating the number of feasible schedules.

Keywords: Scheduling; mixed graph; vertex coloring; chromatic polynomial

Mathematics Subject Classification 1991:
Primary: 90 B 35; Secondary: 05 C 15

1 Introduction

One of the main restrictions for the application of contemporary scheduling theory is that it concentrates on the minimization of a function of the job completion times and does not take into account the cost of using a processing system. In [6, 8], the minimization of a function of both the number of machines used and the job completion times has been considered. The latter problem induces a graph coloring for testing the feasibility of the assignment of jobs to machines and for calculating the value of the objective function. In this paper, we consider a preference relation given on the set of machines and an inadmissibility to assign some operations to the same machine. The former relation may be represented by arcs A and the latter one by edges E in the mixed graph (V, A, E) with vertices V being assumed to be operations. In such a representation, an assignment of operations to the machines may be interpreted as a coloring $\phi : V \rightarrow \{1, 2, \dots, t\}$ of vertices of mixed graph (V, A, E), provided that, if vertices v_i and v_j are joined by an arc $(v_i, v_j) \in A$, then the non-strict

[1]Supported by Deutsche Forschungsgemeinschaft (Project ScheMA), by INTAS (Project 96-0820) and by ISTC (Project B-104)

inequality

$$\phi(v_i) \le \phi(v_j) \tag{1}$$

must hold, and if vertices v_p and v_q are joined by an edge $[v_p, v_q] \in E$, then their colors have to be different:

$$\phi(v_p) \ne \phi(v_q).$$

Two other scheduling examples which also motivate the investigation of mixed graph colorings were given in [3], where a somewhat different definition of coloring has been studied. Namely, instead of the non-strict inequality (1) inequality $\phi(v_i) < \phi(v_j)$ has been required for each arc $(v_i, v_j) \in A$. The first example in [3] has been provided by scheduling a set of jobs with unit processing times subject to both precedence and disjunctive constraints. The second example in [3] is "some type 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". In the framework of the latter example the non-strict inequality (1) allows to consider also the usual condition that some exams (say $v_1, v_2, \ldots, v_k$) have to be taken during the same period since the same tests have to be used in all these exams. To this end, it is sufficient to generate a circuit $(v_1, v_2, \ldots, v_k, v_1)$ in the mixed graph (V, A, E).

These examples and some others (see e.g. [7]) show that a usual coloring of the vertices of a graph $(V, \emptyset, E)$ is too limited to be useful in many scheduling applications. This paper is devoted to a generalization ϕ of the classical coloring model. Both optimization and enumeration questions for a mixed graph coloring are considered. In Section 2 we discuss a rather general scheduling problem for which mixed graph coloring may be considered as the most difficult auxiliary problem. The solution of an illustrative example by a branch-and-bound method is presented in Section 3. In Sections 4 and 5, we present and test algorithms for calculating the chromatic number of a mixed graph. Some properties of the chromatic polynomial of a mixed graph are studied in Section 6. Section 7 contains concluding remarks. The terminology from graph theory used in this paper follows [4], unless defined otherwise.

2 Scheduling jobs on machines of different types

As the main scheduling example which motivates mixed graph coloring, we consider the following problem $\mathcal{P}$ of resource-constrained project scheduling with non-divisable, renewable resources (machines). A set $V = \{v_1, v_2, \ldots, v_n\}$ of non-preemptable operations have to be processed on the set $\mathbf{M} = \{\mu_1, \mu_2, \ldots, \mu_w\}$ of machines. There exist both identical and different machines in the set $\mathbf{M}$, and the assignment of operations to machines is not given in advance. The set $\mathbf{M}$ of machines is subdivided into m mutually disjoint non-empty subsets $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_m$:

$$\mathbf{M} = \cup_{k=1}^{m} \mathbf{M}_k,\ \mathbf{M}_k \ne \emptyset,\ \mathbf{M}_k \cap \mathbf{M}_l = \emptyset,\ k = 1, 2, \ldots, m,\ l = 1, 2, \ldots, m,\ k \ne l,$$

where each set $\mathbf{M}_k$ consists of m_k machines of type k, $|\mathbf{M}_k| = m_k$. The types of machines $\mu_i \in \mathbf{M}_k$ and $\mu_j \in \mathbf{M}_l$ with $k \ne l$ are different. Let V_k denote the set of all operations $v_i \in V$ that have to be processed by a machine of type k, and let the set V be subdivided into m mutually disjoint non-empty subsets $V_1, V_2, \ldots, V_m$:

$$V = \cup_{k=1}^{m} V_k,\ V_k \ne \emptyset,\ V_k \cap V_l = \emptyset,\ k = 1, 2, \ldots, m,\ l = 1, 2, \ldots, m,\ k \ne l.$$

Time constraints for processing operations are assumed to be given by the mixed graph $G = (V, A, E)$ with the vertex (operation) set V, the arc set A and the edge set E. If arc (v_i, v_j) belongs to set A, then the time interval $(c_i - p_i, c_i]$ for the processing of operation $v_i \in V$ has to precede the time interval $(c_j - p_j, c_j]$ for the processing of operation $v_j \in V$, i.e. $c_i \leq c_j - p_j$. Hereafter, p_i and c_i denote the processing time and the completion time of operation v_i, respectively. If edge $[v_i, v_j]$ belongs to set E, then operations v_i and v_j cannot be processed at the same time, i.e. the time intervals $(c_i - p_i, c_i]$ and $(c_j - p_j, c_j]$ may not have common points. The latter constraint is usually presented in the disjunctive form (see e.g. [2, 10]), i.e. one of the following non-strict inequalities must hold:

$$c_i \leq c_j - p_j \text{ or } c_j \leq c_i - p_i. \tag{2}$$

Along with the above *time constraints* given by the mixed graph G, we assume that constraints on the assignment of operations $V_k \subset V$ to the machines M_k are given by digraphs $\Gamma_k = (V_k, A_k, \emptyset)$, $k = 1, 2, \ldots, m$, as follows. If $(v_i, v_j) \in A_k$, then operation v_i (operation v_j) has to be assigned to such a machine $\mu_r \in \mathbf{M}_k$ (to such a machine $\mu_s \in \mathbf{M}_k$), that the non-strict inequality $r \leq s$ holds. These conditions (we shall call them *assignment constraints*) define a preference relation on the set of machines $\mathbf{M}_k$. Assignment constraints may arise in a processing system due to different costs for using different machines of the same type, or due to different precisions with which some operations have to be carried out (provided that in the set $\mathbf{M}_k$ different machines may process an operation with different precisions).

To solve problem $\mathcal{P}$, the following three steps are required:
(C) *choosing* the number $r_k, r_k \leq m_k$, of machines from the set $\mathbf{M}_k, k = 1, 2, \ldots, m$;
(A) *assigning* the operations V_k to the chosen machines, i.e. subdividing the set V_k into r_k mutually disjoint non-empty subsets $V_k^1, V_k^2, \ldots, V_k^{r_k}$;
(S) *sequencing* the operations V according to the above assignment.

All steps (C), (A) and (S) have to be done in an optimal way, i.e. a given non-decreasing objective function $F(c_1, c_2, \ldots, c_n; r_1, r_2, \ldots, r_m)$ must reach its minimal possible value.

While sequencing step (S) (with fixed assignment of operations to machines) is well studied in the OR literature (see e.g. [2, 10]), the choosing step (C) and the assigning step (A) have only limited attention. In particular, in [8] it was shown that steps (C), (A) and (S) may be carried out simultaneously due to the special transformations of the mixed graph G and of the digraphs $\Gamma_k, k = 1, 2, \ldots, m$. Next, we describe this approach briefly (we refer to [8] for a more detailed presentation).

For the assignment of operations $v_i \in V_k$ and $v_j \in V_k$ to the machines from the set $\mathbf{M}_k$ and for sequencing these operations, there exist the following three possibilities.

(*) If the operations v_i and v_j from the set V_k will be processed by the same machine in the order v_j after v_i, then the first non-strict inequality $c_i \leq c_j - p_j$ from (2) must be satisfied.

(**) If the operations v_j and v_i from the set V_k will be processed by the same machine in the opposite order v_i after v_j, then the second non-strict inequality $c_j \leq c_i - p_i$ from (2) must be satisfied.

(***) If the operations v_i and v_j from the set V_k will be processed by different machines from the set $\mathbf{M}_k$, then there is no need to obey the non-strict inequalities (2).

In case (*), the edge $[v_i, v_j] \in E$ is replaced by the arc (v_i, v_j) in the mixed graph $G = (V, A, E)$. In case (**), the edge $[v_i, v_j] \in E$ is replaced by the arc (v_j, v_i) in the mixed graph $G = (V, A, E)$. Moreover, in both cases (*) and (**), the vertices v_i and v_j are identified into one vertex $v = v_i v_j$ in the digraph $\Gamma_k = (V_k, A_k, \emptyset)$. In case (***), the edge $[v_i, v_j]$ is removed from the mixed graph $G = (V, A, E)$, and the same edge $[v_i, v_j]$ is added to the digraph $\Gamma_k = (V_k, A_k, \emptyset)$. Thus, each above transformation of edge $[v_i, v_j] \in E$ in the mixed graph G with $v_i \in V_k$ and $v_j \in V_k$ is accompanied by the corresponding transformation in the digraph Γ_k.

Problem $\mathcal{P}$ may be solved by a suitable sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{|E|})$ of the above transformations of the mixed graph G and of the digraphs Γ_k, $k = 1, 2, \ldots, m$. Let $G[\alpha] = (V, A[\alpha], \emptyset)$ denote the digraph obtained from the mixed graph G as a result of the sequence of transformations α, and let $\Gamma_k[\alpha] = (V_k[\alpha], A_k[\alpha], E_k[\alpha])$, $k = 1, 2, \ldots, m$, denote the mixed graphs obtained from the digraphs Γ_k as a result of the corresponding transformations α. Let $\gamma(\Gamma_k[\alpha])$ denote the *chromatic number* of the mixed graph $\Gamma_k[\alpha]$, i.e. the smallest number of colors t for which there exists coloring $\phi : V_k[\alpha] \rightarrow \{1, 2, \ldots, t\}$. Arguing in a similar way as in [8], we can prove the following claim.

Theorem 1 *The digraph $G[\alpha]$ and the mixed graphs $\Gamma_k[\alpha]$, $k = 1, 2, \ldots, m$, define a feasible schedule for problem $\mathcal{P}$ if and only if the following three conditions hold:*

1) *the digraph $G[\alpha]$ has no circuits;*

2) *the non-strict inequality $\gamma(\Gamma_k[\alpha]) \leq m_k$ holds for each $k = 1, 2, \ldots, m$;*

3) *the mixed graph $\Gamma_k[\alpha]$ has no non-oriented loops for each $k = 1, 2, \ldots, m$.*

Let $R(G)$ and $R(\Gamma_k)$, $k = 1, 2, \ldots, m$, denote the set of all digraphs $G[\alpha]$ and the sets of all mixed graphs $\Gamma_k[\alpha]$, $k = 1, 2, \ldots, m$, for which condition 1, and conditions 2 and 3, respectively, are satisfied. If all conditions 1, 2 and 3 are satisfied, sequence α is called *feasible*. Along with a complete sequence $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{|E|})$ of the transformations of all edges of the mixed graph G, a partial sequence $\alpha' = (\alpha_1, \alpha_2, \ldots, \alpha_l)$, $l < |E|$, will be considered in the framework of a branch-and-bound method. Starting with the one-element sequence $\alpha' = (\alpha_1)$, which transforms one edge, we can construct a feasible complete sequence step by step. Note that, after a complete feasible sequence α, the mixed graph G is transformed into digraph $G[\alpha]$, while digraph Γ_k is transformed into the mixed graph $\Gamma_k[\alpha]$, $k \in \{1, 2, \ldots, m\}$. In other words, after a complete feasible sequence α, the time constraints are defined by digraph $G[\alpha]$, while the assignment constraints are defined by the set of mixed graphs $\Gamma_k[\alpha]$, $k = 1, 2, \ldots, m$. After a partial sequence $\alpha' = (\alpha_1, \alpha_2, \ldots, \alpha_l)$, $l < |E|$, time constraints and assignment constraints are defined by the corresponding mixed graphs.

Let $s_i(V, A[\alpha'], \emptyset)$ and $c_i(V, A[\alpha'], \emptyset)$, respectively, denote the *earliest starting time* and *earliest completion time* of operation $v_i \in V$ in the network $(V, A[\alpha'], \emptyset)$ with the weight p_i being prescribed to each arc $(v_i, v_j) \in A[\alpha']$ with startvertex $v_i \in V$, i.e. $s_i(V, A[\alpha'], \emptyset)$ is equal to the maximal weight of a path in network $(V, A[\alpha'], \emptyset)$ ending in vertex $v_i \in V$, and $c_i(V, A[\alpha'], \emptyset) = s_i(V, A[\alpha'], \emptyset) + p_i$.

The edge $[v_i, v_j] \in E$ is called a *conflict edge* (see e.g. [10]) if both inequalities

$$s_j(V, A[\alpha'], \emptyset) < c_i(V, A[\alpha'], \emptyset) \tag{3}$$

$$s_i(V, A[\alpha'], \emptyset) < c_j(V, A[\alpha'], \emptyset) \tag{4}$$

hold. Obviously, if we replace conflict edge $[v_i, v_j]$ by the arc (v_i, v_j), then due to inequality (3), the starting time and completion time of operation v_j will be increased. If we replace conflict edge $[v_i, v_j]$ by the arc (v_j, v_i), then due to inequality (4), the starting time and completion time of operation v_i will be increased. On the other hand, if edge $[v_i, v_j]$ is not a conflict one, then there exists an orientation of this edge (say (v_i, v_j)) such that neither starting times nor completion times will be increased in the network $(V, A[\alpha'] \cup \{(v_i, v_j)\}, \emptyset)$.

By $G|v_iv_j$ we denote the mixed graph which is obtained from $G = (V, A, E)$ by identifying the two vertices $v_i \in V$ and $v_j \in V$. More precisely, by $G|v_iv_j$ we denote the mixed graph obtained from G by deleting both vertices v_i and v_j, by deleting all the edges and arcs incident to v_i or v_j, by introducing a new vertex $v = v_iv_j$, by joining by arcs (by edges) vertex v to all the neighbors of v_i and v_j in digraph $(V, A, \emptyset)$ (in graph $(V, \emptyset, E)$, respectively), and by replacing all resulting multiple edges (multiple arcs) by a single edge (by a single arc, respectively). Note that, if there exists the arc (v_i, v_j) (the edge $[v_i, v_j]$) in the mixed graph G, then there exists the oriented loop (v, v) (the non-oriented loop $[v, v]$, respectively) in the mixed graph $G|v_iv_j$.

Next, we briefly describe a branch-and-bound algorithm for problem $\mathcal{P}$. Let $[v_i, v_j]$ be a conflict edge in the mixed graph G with $v_i \in V_k$ and $v_j \in V_k$. At the first iteration, we transform edge $[v_i, v_j]$ in the three possible ways (*), (**) and (***). As a result, we obtain a partition of the set $R(G)$ into three subsets: $R(G^1) = R(V, A \cup \{(v_i, v_j)\}, E \setminus \{[v_i, v_j]\})$, $R(G^2) = R(V, A \cup \{(v_j, v_i)\}, E \setminus \{[v_i, v_j]\})$ and $R(G^3) = R(V, A, E \setminus \{[v_i, v_j]\})$. In turn, set $R(\Gamma_k)$ is partitioned into three corresponding subsets $R(\Gamma_k^1) = R(\Gamma_k|v_iv_j)$, $R(\Gamma_k^2) = R(\Gamma_k|v_jv_i)$ and $R(\Gamma_k^3) = R(V_k, A_k, \{[v_i, v_j]\})$, where $\Gamma_k^1 = \Gamma_k^2$. At the second iteration, we take one of the mixed graphs G^1, G^2 or G^3 and continue with it in a similar way as with the mixed graph G at the first iteration, and so on. At each iteration, we test condition 1 from Theorem 1 for the mixed graph $G^l = (V, A[\alpha'], E[\alpha'])$ obtained and conditions 2 and 3 for the mixed digraph $\Gamma_k^l = \Gamma_k[\alpha'] = (V_k[\alpha'], A_k[\alpha'], E_k[\alpha'])$ obtained. If all of these conditions are satisfied and the mixed graph G^l contains at least one conflict edge, then a lower bound of the objective function F may be calculated on the basis of the following non-strict inequality (see [8]):

$$\begin{aligned} &F(c_1(V, A[\alpha'], \emptyset), c_2(V, A[\alpha'], \emptyset), \ldots, c_n(V, A[\alpha'], \emptyset); \gamma(\Gamma_1[\alpha']), \gamma(\Gamma_2[\alpha']), \ldots, \gamma(\Gamma_m[\alpha'])) \quad (5)\\ &\le F(c_1(V, A[\alpha], \emptyset), c_2(V, A[\alpha], \emptyset), \ldots, c_n(V, A[\alpha], \emptyset); \gamma(\Gamma_1[\alpha]), \gamma(\Gamma_2[\alpha]), \ldots, \gamma(\Gamma_m[\alpha])), \end{aligned}$$

which holds for any complete feasible sequence $\alpha = (\alpha', \alpha'')$ constructed from the partial feasible sequence α'.

If there is no conflict edge in the mixed graph $G^l = (V, A[\alpha'], E[\alpha'])$, then the non-strict inequality (5) holds as equality for some complete feasible sequence $\alpha^* = \alpha = (\alpha', \alpha'')$. Thus, the mixed graph G^l and the mixed graphs $\Gamma_1^l, \Gamma_2^l, \ldots, \Gamma_m^l$ define a feasible solution $(V, A[\alpha^*], \emptyset)$, $\Gamma_1[\alpha^*], \Gamma_2[\alpha^*], \ldots, \Gamma_m[\alpha^*]$ to problem $\mathcal{P}$ with the value of the objective function being equal to $F(c_1(V, A[\alpha^*], \emptyset), c_2(V, A[\alpha^*], \emptyset), \ldots, c_n(V, A[\alpha^*], \emptyset); \gamma(\Gamma_1[\alpha^*]), \gamma(\Gamma_2[\alpha^*]), \ldots, \gamma(\Gamma_m[\alpha^*]))$. The latter is used as stopping rule in the branch-and-bound algorithm.

Instead of giving the background of such a mixed graph approach to problem $\mathcal{P}$, we shall illustrate it on a small example in Section 3 and we refer to [8], where the corresponding branch-and-bound algorithm has been substantiated. Note that the assignment constraints, which were not considered in [8], do not cause essential differences in the proofs of the necessary claims. The only exception is connected with mixed graph coloring, since in the scheduling problem considered in [8] a classical graph coloring was sufficient for the verification of an analogue of condition 2 in Theorem 1.

It should be noted that condition 1 (and condition 3) may be verified in linear time in the number of arcs $A[\alpha]$ (in the number of edges $E_k[\alpha]$, respectively), while the verification of condition 2 is an NP-complete problem, since it is already an NP-complete problem even if $A_k[\alpha] = \emptyset$ (see [5]). So, the rest of this paper is focused on mixed graph coloring as the most difficult auxiliary problem for $\mathcal{P}$. Before presenting an illustrative example of problem $\mathcal{P}$ in Section 3 and the algorithms for mixed graph coloring in Section 4, we give two remarks.

Remark 1 *For the verification of condition 2 in Theorem 1, it may be sufficient to calculate an upper bound of the chromatic number of a mixed graph.*

Indeed, to test condition 2 in Theorem 1, we do not need to define the exact value $\gamma(\Gamma_k[\alpha])$, but we must solve only the following decision problem. Given a positive integer m_k, is there a coloring of the mixed graph $\Gamma_k[\alpha]$ with no more than m_k colors used?

The next remark follows from the extraordinary complexity of problem $\mathcal{P}$. Even after the realization of steps (C) and (A) in problem $\mathcal{P}$, we obtain a job-shop problem which is strongly NP-hard even for three machines ($r_k = 3$) and which can be solved practically only for a very small number r_k (see e.g. [2]). In particular, in the computational experiments performed in [8], the largest number r_k of machines for which problem $\mathcal{P}$ was solved exactly was restricted by 8.

Remark 2 *Due to the notorious complexity of problem $\mathcal{P}$, it is sufficient to consider mixed graphs with small chromatic numbers.*

Our selection of algorithms for calculating the chromatic number will be based on both above remarks.

3 Example

In this section, we illustrate the above approach to problem $\mathcal{P}$ on a small example. Let the processing system be composed of machines of two types $M_1 = \{\mu_1, \mu_2\}$ and $M_2 = \{\mu_3, \mu_4\}$, and assume that it has to process 6 operations: $V = \{v_1, v_2, \dots, v_6\}$. The set of operations is subdivided into two subsets according to the relevant machine types: $V_1 = \{v_1, v_2, v_4\}, V_2 = \{v_3, v_5, v_6\}$. The operation processing times are as follows: $p_1 = 3, p_2 = 4, p_3 = 6, p_4 = 5, p_5 = 7, p_6 = 5$. Time constraints are given by the mixed graph $G = (V, A, E)$ presented in Figure 1a.

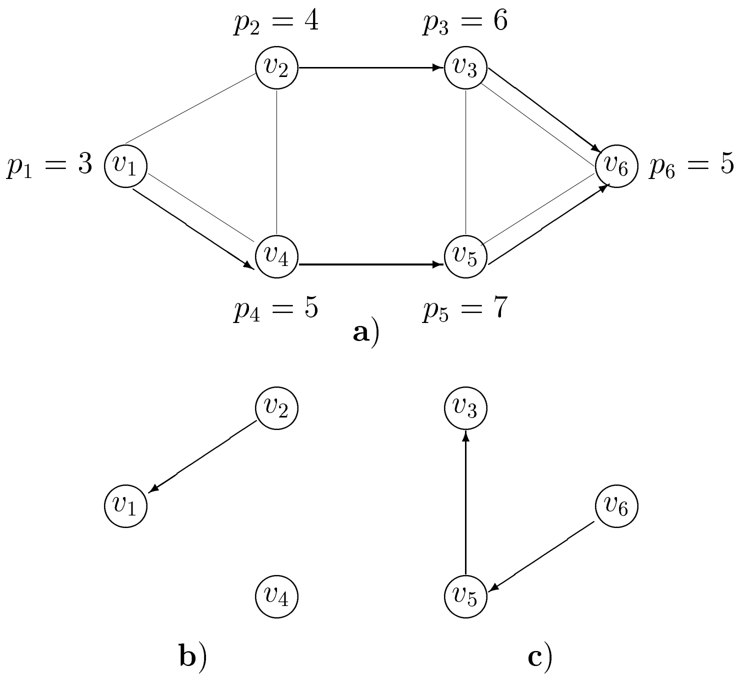

FIGURE 1. a) Mixed graph $G = (V, A, E)$,
b) digraph $\Gamma_1 = (V_1, A_1, \emptyset)$,
c) digraph $\Gamma_2 = (V_2, A_2, \emptyset)$.

Constraints on the assignment of operations V_1 and V_2 to the machines of the sets M_1 and M_2 (i.e. assignment constraints) are given by the digraphs $\Gamma_1 = (V_1, A_1, \emptyset)$ and $\Gamma_2 = (V_2, A_2, \emptyset)$ (see Figure 1b and Figure 1c, respectively). The objective is to minimize function $F(c_1, c_2, \dots, c_6; r_1, r_2) = \max\{c_i : v_i \in V\} + r_1^2 + r_2$.

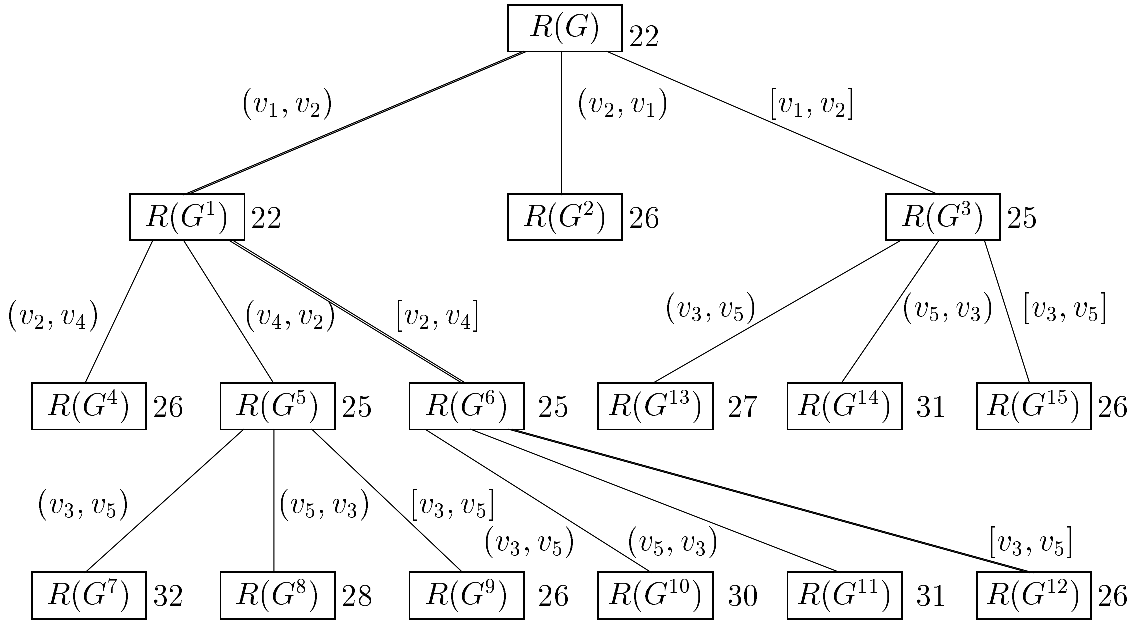

FIGURE 2. Solution tree.

The solution tree for this example obtained due to the branch-and-bound algorithm is presented in Figure 2. The corresponding sets $R(G^k)$ are vertices of this tree. Either the arc (v_i, v_j) or the arc (v_j, v_i) or the edge $[v_i, v_j]$ is prescribed to arc $(R(G^k), R(G^l))$ of the solution tree depending on whether the mixed graph G^l is obtained from G^k by replacing the edge $[v_i, v_j]$ by the arc (v_i, v_j) or by the arc (v_j, v_i) or by removing the edge $[v_i, v_j]$. The lower bound of the objective function $F(c_1, c_2, \ldots, c_6; r_1, r_2)$ calculated according to (5) is assigned on the right to each set $R(G^l)$ in Figure 2.

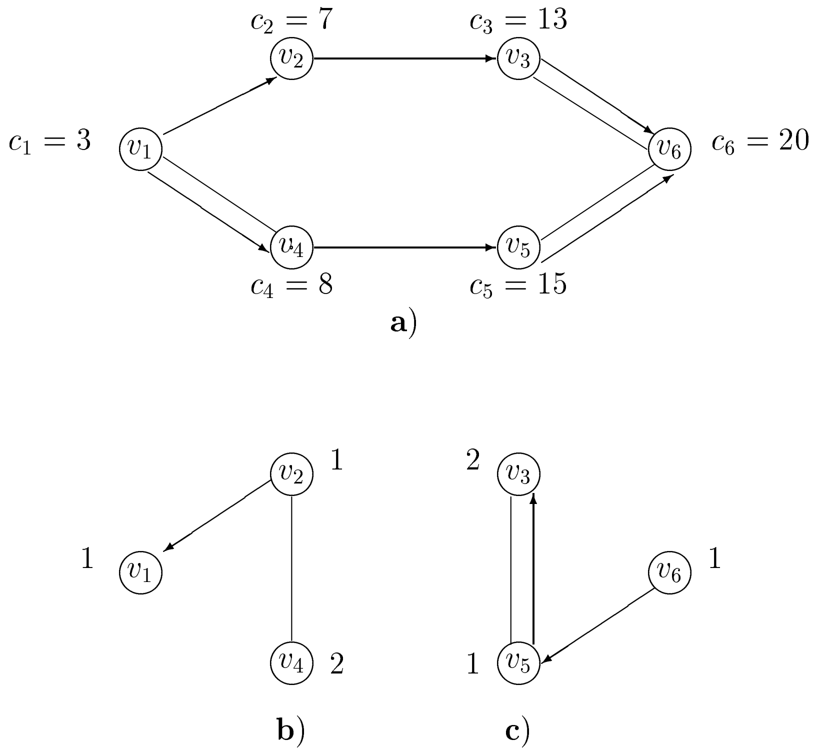

FIGURE 3. a) Mixed graph $G^{12} = (V, A[\alpha], E[\alpha])$,
b) mixed graph $\Gamma_1^{12} = (V_1[\alpha], A_1[\alpha], E_1[\alpha])$,
c) mixed graph $\Gamma_2^{12} = (V_2[\alpha], A_2[\alpha], E_2[\alpha])$.

The mixed graph G^{12} represented in Figure 3a contains no circuits (see condition 1 in Theorem 1) and no conflict edges. The corresponding mixed graphs Γ_1^{12} (see Figure 3b) and Γ_2^{12} (see Figure 3c) satisfy both conditions 2 and 3 of Theorem 1. Thus, the mixed graphs G^{12}, Γ_1^{12} and Γ_2^{12} define a set of feasible solutions to problem $\mathcal{P}$. Obviously, the chromatic numbers for both mixed graphs Γ_1^{12} and Γ_2^{12} are equal to 2 (optimal colorings are presented in Figures 3b and 3c). Therefore, the best feasible solution defined by mixed graphs G^{12}, Γ_1^{12} and Γ_2^{12} is as follows: $r_1 = r_2 = 2, V_1^1 = \{v_1, v_2\}, V_1^2 = \{v_4\}, V_2^1 = \{v_5, v_6\}, V_2^2 = \{v_3\}, c_1 = 3, c_2 = 7, c_3 = 13, c_4 = 8, c_5 = 15, c_6 = 20$. The value of the objective function for this solution is equal to

$$F(c_1(G^{12}), c_2(G^{12}), \ldots, c_6(G^{12}); \gamma(\Gamma_1^{12}), \gamma(\Gamma_2^{12})) = \max\{3, 7, 13, 8, 15, 20\} + 2^2 + 2 = 26.$$

Since this value is equal to or less than the lower bounds of the values of the objective function for all other sets of solutions $R(G^4), R(G^7), R(G^8), R(G^9), R(G^{10}), R(G^{11}), R(G^{13}), R(G^{14}), R(G^{15})$

which together with set $R(G^{12})$ form a partition of the set $R(G)$ (see Figure 2), the above best feasible solution defined by the mixed graphs G^{12}, Γ_1^{12} and Γ_2^{12} is optimal.

In Figure 2, the path from vertex $R(G)$ to vertex $R(G^{12})$, drawn in bold face, defines the partial sequence $\alpha' = (\alpha_1, \alpha_2, \alpha_3)$, where α_1 means replacing edge $[v_1, v_2]$ by arc (v_1, v_2) in the mixed graph G, α_2 means removing edge $[v_2, v_4]$ from the mixed graph G^1, and α_3 means removing edge $[v_3, v_5]$ from the mixed graph G^6.

4 The calculation of the chromatic number

The mixed graph approach to the scheduling problem $\mathcal{P}$ described and illustrated in Sections 2 and 3 implies the calculation of the chromatic number of a mixed graph or at least its upper bound for testing the assignment constraints (see condition 2 in Theorem 1) and for calculating a lower bound or the exact value of the objective function $F(c_1, c_2, \ldots, c_n; r_1, r_2, \ldots, r_m)$ (see (5)). In this section, after introducing additional notations, we prove a criterion for the existence of a mixed graph coloring and describe three algorithms for constructing an optimal coloring or a coloring with a rather small number of colors used.

Let $G = (V, A, E)$ be a mixed pseudograph with the vertex set $V = \{v_1, v_2, \ldots, v_n\}$, the arc set A and the edge set E. More precisely, the mixed pseudograph G under consideration may contain loops (either oriented or non-oriented), but no multiple edges or multiple arcs. However, we shall continue to call G a mixed graph (instead of a mixed pseudograph).

Definition 1 *We denote a function ϕ as a coloring of the mixed graph G, if it assigns to each vertex v_k, $k = 1, 2, \ldots, n$, a positive integer $\phi(v_k)$ such that $\phi(v_i) \le \phi(v_j)$ if $(v_i, v_j) \in A$ and $\phi(v_p) \ne \phi(v_q)$ if $[v_p, v_q] \in E$. The number of different functions ϕ with values from the set $\{1, 2, \ldots, t\}$, which are colorings of the mixed graph G, is denoted by $f(G, t)$. The smallest number t for which $f(G, t) > 0$ is called the chromatic number of the mixed graph G, and it is denoted by $\gamma(G)$. The coloring $\phi : V \to \{1, 2, \ldots, \gamma(G)\}$ of the mixed graph G is optimal.*

First, we derive some properties of coloring ϕ, number $\gamma(G)$ and function $f(G, t)$. If $A = \emptyset$, then function ϕ from Definition 1 gives a usual coloring of graph $(V, \emptyset, E)$, $\gamma(G)$ is the chromatic number of graph $G = (V, \emptyset, E)$, and $f(G, t)$ is the chromatic polynomial of graph $G = (V, \emptyset, E)$ (see e.g. [1, 4, 11]). If $E = \emptyset$, then $f(G, t)$ is the chromatic polynomial of digraph $G = (V, A, \emptyset)$ (see e.g. [9]). As it will be shown, the most difficult part for calculating $\gamma(G)$ for the mixed graph $G = (V, A, E)$ is accounting the edges $[v_i, v_j] \in E$ with vertices v_i and v_j being not adjacent in the digraph $(V, A, \emptyset)$.

Obviously, if $A = \emptyset$, then there exists a coloring of graph $G = (V, \emptyset, E)$, e.g. $\phi(v_i) = i$ for each vertex $v_i \in V$. If $E = \emptyset$, then the constant function $\phi(v_i) = 1$, $v_i \in V$, is an optimal coloring of the digraph $G = (V, A, \emptyset)$, and thus $\gamma(V, A, \emptyset) = 1$. However, if both sets A and E are not empty: $A \ne \emptyset$ and $E \ne \emptyset$, then there may not exist a coloring ϕ of the mixed graph $G = (V, A, E)$. For example, if there exists a non-oriented loop $[v_i, v_i] \in E$, then there does not exist a coloring of the mixed graph $G = (V, A, E)$ since due to Definition 1, for each coloring ϕ the existence of loop $[v_i, v_i] \in E$ implies a contradiction: $\phi(v_i) \ne \phi(v_i)$. Also, there does not exist a coloring of the mixed graph G if its directed subgraph $(V, A, \emptyset)$ contains a circuit $\mu = (v_k, \ldots, v_i, \ldots, v_j, \ldots, v_k)$ with the vertices v_i and v_j for which inclusion $[v_i, v_j] \in E$ holds. Indeed, due to circuit μ for each coloring ϕ, we have the non-strict inequalities $\phi(v_k) \leq \ldots \leq \phi(v_i) \leq \ldots \leq \phi(v_j) \leq \ldots \leq \phi(v_k)$ which imply the equality $\phi(v_i) = \phi(v_j)$. But the existence of edge $[v_i, v_j]$ implies a contradiction: $\phi(v_i) \neq \phi(v_j)$.

Next, we prove that the above two cases are the only possible ones when a coloring ϕ does not exist for a mixed graph G. To this end, we generalize notation $G|v_iv_j$ introduced in Section 2 on the set N of pairs of vertices: by $G|N = (V|N, A|N, E|N)$ we denote the mixed graph which is obtained from G by identifying consecutively the vertices v_i and v_j for each pair $\{v_i, v_j\} \in N$. If $N = \emptyset$, we set $G|N = G$.

Assume now that the mixed graph $G = (V, A, E)$ does not contain non-oriented loops and circuits μ described above. If the set of arcs $L = \{(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots, (v_{i_{s-1}}, v_{i_s}), (v_{i_s}, v_{i_1})\}$ forms a circuit $\nu = (v_{i_1}, v_{i_2}, \ldots, v_{i_s}, v_{i_1})$ in the digraph $(V, A, \emptyset)$ without the vertices that are adjacent in graph $(V, \emptyset, E)$, then we can consider the mixed graph $G|L$. There is a one-to-one correspondence between the set of colorings ϕ of the original mixed graph G and the set of colorings $\tilde{\phi}$ of the mixed graph $G|L$. Indeed, we set $\phi(v_{i_j}) = \tilde{\phi}(v)$ for all vertices v_{i_j} which are contained in circuit ν and we set $\phi(v_k) = \tilde{\phi}(v_k)$ for all remaining vertices v_k from set V. Here v is a vertex of the mixed graph $G|L$ into which all vertices contained in circuit ν have been identified. In particular, it follows from this one-to-one correspondence that

$$f(G, t) = f(G|L, t) \text{ and } \gamma(G) = \gamma(G|L).$$

By repeating the above procedure for each such circuit ν being contained in digraph $(V, A, \emptyset)$, we obtain a mixed graph with a circuit-free directed subgraph and with the same chromatic number as the original mixed graph G has.

Let the mixed graph $G_k^E = (V, A \cup A_k^E, E)$ with the circuit-free directed subgraph $(V, A \cup A_k^E, \emptyset)$ be called an *extension* of the mixed graph $G = (V, A, E)$, if it is obtained from G as a result of adding one of the arcs (v_i, v_j) or (v_j, v_i) for each pair of vertices v_i and v_j which are adjacent in graph $(V, \emptyset, E)$ and not adjacent in digraph $(V, A, \emptyset)$. Similarly to [10, p. 286], it is easy to prove the following claim.

Lemma 1 *Let graph $(V, \emptyset, E)$ have no loops. Then an extension of the mixed graph $G = (V, A, E)$ exists if and only if digraph $(V, A, \emptyset)$ has no circuits.*

Proof: If digraph $(V, A, \emptyset)$ has no circuit, then there exist an ordering of vertices

$$V = (v^{(1)}, v^{(2)}, \ldots, v^{(n)})$$

such that inclusion $(v^{(i)}, v^{(j)}) \in E$ implies inequality $i < j$. This ordering defines the desired extension G_k^E of the mixed graph G as follows. We include arc $(v^{(p)}, v^{(q)})$ into the set A_k^E if and only if $[v^{(p)}, v^{(q)}] \in E$, $p < q$ and vertices $v^{(p)}$ and $v^{(q)}$ are not adjacent in digraph $(V, A, \emptyset)$.

On the other hand, if digraph $(V, A, \emptyset)$ has a circuit, then any digraph $(V, A \cup A', \emptyset)$ has this circuit, and so no extension of the mixed graph G exists. ∎

Since G is a subgraph of G_k^E, each coloring of the mixed graph G_k^E is a coloring of the mixed graph G, too. It is important that an optimal coloring of the mixed graph G_k^E may be calculated by the critical path method (CPM). To this end, we prescribe weight 0 to each arc $(v^{(i)}, v^{(j)}) \in A$ with $[v^{(i)}, v^{(j)}] \notin E$, and weight 1 to each arc $(v^{(p)}, v^{(q)}) \in A \cup A_k^E$ with $[v^{(p)}, v^{(q)}] \in E$. By CPM, one can calculate the earliest starting time $s_i(V, A \cup A_k^E, E)$ for each vertex $v_i \in V$ in the network $(V, A \cup A_k^E, E)$. It is easy to convince that function $\phi(v_i) = s_i(V, A \cup A_k^E, E)$, $v_i \in V$, is an optimal coloring for the mixed graph G_k^E and therefore $\gamma(G_k^E)$ is equal to the *critical weight*, i.e. to the maximal weight of a directed path in the weighted mixed graph $(V, A \cup A_k^E, E)$:

$$\gamma(G_k^E) = \max\{s_i(V, A \cup A_k^E, E) \ : \ v_i \in V\}.$$

In particular, we obtain the following claim.

Theorem 2 *For the existence of a coloring of the mixed graph $G = (V, A, E)$, it is necessary and sufficient that graph $(V, \emptyset, E)$ has no loops and digraph $(V, A, \emptyset)$ does not contain any circuit with adjacent vertices of graph $(V, \emptyset, E)$.*

Let the mixed graph G be called *k-colorable*, if there exists a coloring ϕ for the mixed graph G with no more than k colors. As a consequence of Theorem 2, we can give the following simpler formulation of Theorem 1.

Theorem 1' *The digraph $G[\alpha]$ and the mixed graphs $\Gamma_k[\alpha], k = 1, 2, \dots, m$, define a feasible schedule for problem $\mathcal{P}$ if and only if the following two conditions hold:*

1) the digraph $G[\alpha]$ has no circuits;

2) the mixed graph $\Gamma_k[\alpha]$ is m_k-colorable for each $k = 1, 2, \dots, m$.

From the above proof of Theorem 2, it follows that, to find an optimal coloring of the mixed graph G, it is sufficient to find the best extension G_k^E of the mixed graph G, i.e. such a set of arcs A_k^E that the weighted mixed graph $(V, A \cup A_k^E, E)$ has a minimal critical weight. The calculation of an optimal coloring for a mixed graph G_k^E by CPM takes $O(|A \cup A_k^E|)$ time.

Obviously, value $\gamma(G_k^E)$ calculated for any extension G_k^E of the mixed graph G provides an upper bound of $\gamma(G)$. A lower bound of $\gamma(G)$ may also be calculated by CPM applied to the mixed graph $G^A = (V, A, E \setminus E^A)$ obtained from G as a result of the deletion of all edges $[v_i, v_j] \in E^A \subseteq E$ for which vertices v_i and v_j are not adjacent in digraph $(V, A, \emptyset)$. Thus, the following lower and upper bounds of $\gamma(G)$ may be calculated by CPM in $O(n^2)$ time.

Corollary 1 *If graph $(V, \emptyset, E)$ has no loops and digraph $(V, A, \emptyset)$ has no circuits, then*

$$\max\{s_i(V, A, E \setminus E^A) : v_i \in V\} \leq \gamma(V, A, E) \leq \max\{s_i(V, A \cup A_k^E, E) : v_i \in V\}. \quad (6)$$

Note that there exists a unique mixed graph G^A defined by the mixed graph G, while there exist a set of different extensions $G_1^E, G_2^E, \dots, G_k^E, \dots, G_\lambda^E$ of the mixed graph G and the cardinality λ of this set being restricted by $2^{|A_k^E|}$. Next, we show that both lower and upper bounds presented by (6) are tight. Indeed, $\gamma(V, A, E) = \gamma(G_k^E)$ if and only if

$$\gamma(G_k^E) = \min\{\gamma(G_1^E), \gamma(G_2^E), \dots, \gamma(G_\lambda^E)\}.$$

On the other hand, $\gamma(V, A, E) = \gamma(G^A)$ if there is no conflict edge in the set of edges E of the mixed graph G. Obviously, a conflict edge of the mixed graph G (if any) has to belong to set E^A and in this case conditions (3) and (4) of the conflictness of edge $[v_i, v_j] \in E^A$ may be presented as follows: the edge $[v_i, v_j]$ is a conflict one if and only if

$$s_i(V, A, E \setminus E^A) = s_j(V, A, E \setminus E^A).$$

Note that in [3], other upper bounds have been derived for the mixed graph $G = (V, A, E)$ which has the following

Property 1: Inclusion $(v_i, v_j) \in A$ implies inclusion $[v_i, v_j] \in E$.

Those bounds are based on the calculation of the usual chromatic number $\gamma(V, \emptyset, E)$, which is an NP-hard problem.

Let $(V, \tilde{A}, \emptyset)$ be the digraph obtained from $(V, A, \emptyset)$ by reversing the *orientation* of all arcs (i.e. inclusion $(v_i, v_j) \in \tilde{A}$ holds if and only if $(v_j, v_i) \in A$), and let ϕ be an arbitrary coloring of the mixed graph G with colors from the set $\{1, 2, \ldots, t\}$. It is easy to see that the function $\tilde{\phi}(v_i) = t - \phi(v_i) + 1,\ i = 1, 2, \ldots, n$, is a coloring of the mixed graph $(V, \tilde{A}, E)$. Since the converse argument is also true, we get the following equalities:

$$f(G, t) = f((V, \tilde{A}, E), t) \text{ and } \gamma(G) = \gamma(V, \tilde{A}, E).$$

We now consider the problem of determining an optimal coloring of an arbitrary given mixed graph $G = (V, A, E)$ with the circuit-free digraph $(V, A, \emptyset)$ and with the graph $(V, \emptyset, E)$ which has no loops. We describe an algorithm for the latter problem which contains the following 6 main steps. In Step 1, there is determined a list of immediate predecessors P_i of each vertex $v_i \in V$ in the digraph $(V, A, \emptyset)$. In Step 2, if digraph $(V, A, \emptyset)$ has no circuits, the vertices V are ordered and indexed as in the proof of Lemma 1: $V = \{v^{(1)}, v^{(2)}, \ldots, v^{(n)}\}$. If digraph $(V, A, \emptyset)$ has a circuit, the algorithm stops. In Step 3, for each index $k = 1, 2, \ldots, n$, the list S_k is determined which contains all vertices $v^{(i)}$ that are adjacent to vertex $v^{(k)}$ in the graph $(V, \emptyset, E)$ and inequality $i < k$ holds:

$$S_k = \{v^{(i)} \ :\ [v^{(i)}, v^{(k)}] \in E,\ i < k\}.$$

In Step 4, the vertices $v^{(k)}, k = 1, 2, \ldots, n$, are colored greedily according to non-decreasing superscripts k by taking into account the already colored vertices from the lists P_k and S_k. Steps 1 - 4 form a heuristic greedy algorithm, which generates a coloring ϕ_1 with a rather small number of colors used: the color $\phi_1(v^{(k)})$ assigned to vertex $v^{(k)} \in V$ is the smallest positive integer which does not contradict to the colors of the vertices $v^{(1)}, v^{(2)}, \ldots, v^{(k-1)}$. An optimal coloring is constructed by an enumeration of colorings which are no worse than coloring ϕ_1. To this end, Steps 4, 5 and 6 are realized repeatedly, where vertex $v^{(d)}$ is selected whose increase of the current color may lead to a decrease of the total number of colors used. After increasing the color of vertex $v^{(d)}$ at least by 1: $\phi_2(v^{(d)}) \geq \phi_1(v^{(d)}) + 1$, a new coloring ϕ_2 is constructed at Step 4, where the vertices $v^{(j)} \in V$ are colored again if $j > d$. All such vertices $v^{(d)}$ and all possible colors $\phi_r(v^{(d)})$ are considered in order to enumerate (explicitely or implicitely) all colorings which may be better than coloring ϕ_1.

The input data for this algorithm (we call it Algorithm A&AE&E) are given by a list (an ordered set) of the vertices V, a list of the arcs A and a list of the edges E. Along with the sets and parameters defined in the previous paragraph, we shall also use the following ones

in the formal description of Algorithm A&AE&E:
C is the set of vertices which are candidates for increasing the current color;
h is the superscript of vertex $v^{(h)}$ with the maximal color used in the coloring ϕ_r under consideration;
l is the new color of vertex $v^{(d)}$, with which the recoloring of vertices $v^{(d)}, v^{(d+1)}, \dots, v^{(n)}$ starts.

Algorithm A&AE&E

Step 1. **Construction of the lists P_i of immediate predecessors for all vertices $v_i \in V$.**

1a) Considering the set of arcs A, determine the cardinality of the list P_i for each vertex $v_i \in V$, and determine the list of indicators $w(i)$ of the positions of the ends of the lists P_i in the common list $P = (P_1, P_2, \dots, P_n)$:

$$\begin{aligned} w(1) &= |P_1|, \\ w(i+1) &= w(i) + |P_{i+1}|, \quad i = 1, 2, \dots, n-1. \end{aligned}$$

1b) Determine list P in the second run through the arc set A.

Step 2. **Determination of an ordering $V = \{v^{(1)}, v^{(2)}, \dots, v^{(n)}\}$ of the vertices such that from inclusion $(v^{(i)}, v^{(j)}) \in A$ inequality $i < j$ follows.**

2a) Considering the list of arcs A, determine the set V^* of all vertices which do not have a successor, and assign to them the superscripts $n, n-1, \dots, n-u+1$, where $u = |V^*|$:

$$V^* = \{v^{(n-u+1)}, v^{(n-u+2)}, \dots, v^{(n-1)}, v^{(n)}\}.$$

If $V^* = \emptyset$ and there are vertices in the set V which have no desired superscripts, then **STOP: digraph $(V, A, \emptyset)$ has a circuit.**
Otherwise go to Step 2b.

2b) Eliminate arcs $A^* = \{(v_i, v_j) \ : v_j \in V^*\}$ from set A and eliminate vertices V^* from set V.
If $u < n$, then go to Step 2a provided that $V := V \setminus V^*$, $A := A \setminus A^*$ and $n := n - u$.
Otherwise, the desired superscripts have been already assigned to all vertices V.

Step 3. **Construction of the lists S_k for all vertices $v^{(k)} \in V$.**

3a) Considering the list of edges E, determine the cardinality of the list S_k for each vertex $v^{(k)} \in V$, and determine the list of indicators $u(k)$ of the ends of the lists S_k in the common list $S = (S_1, S_2, \dots, S_n)$:

$$\begin{aligned} u(1) &= |S_1|, \\ u(k+1) &= u(k) + |S_{k+1}|, \quad k = 1, 2, \dots, n-1. \end{aligned}$$

3b) Determine list S by the second run through the edge set E.

3c) Set $r = d = l = 1$ and $q = +\infty$.

Step 4. **Coloring of the vertices $\{v^{(d)}, v^{(d+1)}, \dots, v^{(n)}\} \subseteq V$.**

4a) Set $\phi_r(v^{(d)}) = l$ and $k = d + 1$.

4b) Set the color of the vertex $v^{(k)}$ to be equal to the smallest positive integer $\phi_r(v^{(k)})$ from the set $\{1, 2, \dots, k\}$ for which the following conditions are satisfied:

$$\text{if } v^{(i)} \in S_k, \text{ then } \phi_r(v^{(k)}) \neq \phi_r(v^{(i)});$$

$$\text{if } v^{(j)} \in P_u \text{ and } v_u = v^{(k)}, \text{ then } \phi_r(v^{(k)}) \geq \phi_r(v^{(j)}).$$

4c) If $\phi_r(v^{(k)}) \geq q$, then set $h = k$ and go to Step 6.
Otherwise go to Step 4d.

4d) Set $k := k + 1$. If $k \leq n$, then go to Step 4b, otherwise go to Step 5.

Step 5. **Changing the best coloring ϕ^* constructed.**

5a) Set $C = \emptyset$. Set $\phi^*(v^{(k)}) = \phi_r(v^{(k)}),\ k = d, d + 1, \dots, n$.

5b) Select vertex $v^{(h)}$ with the minimal index h having the maximal color $\phi^*(v^{(h)}) = \max_{1 \leq i \leq n} \phi^*(v^{(i)})$. If $q = +\infty$, then set $q_1 = \phi^*(v^{(h)})$.

5c) Set $q = \phi^*(v^{(h)})$.

Step 6. **Selection of the vertex from set C with the maximal superscript which permits to increase its current color.**

6a) Modify the set C of vertices which are candidates for increasing the current color: $C := C \cup S_h \cup P_u$, where $v_u = v^{(h)}$.

6b) If $C = \emptyset$, then **STOP: the currently best coloring ϕ^* is optimal.**
Otherwise select vertex $v^{(t)} \in C$ which has the maximal superscript t among all vertices in the set C.

6c) Determine the smallest possible color $s \geq \phi_r(v^{(t)}) + 1$ which is feasible for vertex $v^{(t)}$ taking into account the colors $\phi_r(v^{(i)})$ of the vertices $v^{(i)} \in S_t$ (i.e. if $v^{(i)} \in S_t$, then $s \neq \phi_r(v^{(i)})$).

6d) If $s \geq q$, then eliminate vertex $v^{(t)}$ from the set C and include vertices S_t and vertices P_u, $v_u = v^{(t)}$, into the set C, i.e.

$$C := C \setminus \{v^{(t)}\} \cup S_t \cup P_u.$$

Go to Step 6b.
If $s < q$, then set $r := r + 1$, $d = t$, $l = s$ and go to Step 4.

The correctness of Algorithm A&AE&E follows from

Lemma 2 *If Algorithm A&AE&E stops at Step 6b, then coloring ϕ^* is optimal.*

Proof: First, we note that each function ϕ_r, constructed at Step 4, Step 5a and Step 6c, is a coloring of the mixed graph G. Indeed, at Steps 4a, 4b, 5a and 6a, color $\phi_r(v^{(k)})$ assigned to vertex $v^{(k)}$, $k \in \{2, 3, \dots, n\}$, does not contradict to the colors $\phi_r(v^{(1)}), \phi_r(v^{(2)}), \dots, \phi_r(v^{(k-1)})$ of all vertices $v^{(1)}, v^{(2)}, \dots, v^{(k-1)}$, which have been colored in coloring ϕ_r before vertex $v^{(k)}$. Note also that color s used at Step 6c does not contradict to color $\phi_r(v^{(i)})$ of vertex $v^{(i)} \in P_u$, $v_u = v^{(t)}$, since s is greater than $\phi_r(v^{(t)})$.

Secondly, let us show that all colorings $\phi_1, \phi_2, \dots, \phi_{r^*}$ are different, where ϕ_{r^*} is the last coloring constructed by Algorithm A&AE&E. Indeed, for each coloring ϕ_r with $r \in \{2, 3, \dots, r^*\}$ considered at Steps 5 and 6 there exist vertex $v^{(d)}$ which has a greater color than that in coloring ϕ_{r-1} (see Step 6c). Thus, coloring ϕ_r differs from all colorings $\phi_1, \phi_2, \dots, \phi_{r-1}$ which were considered at Steps 5 and 6 before coloring ϕ_r.

To finish the proof of Lemma 2, we note that at Steps 4, 5 and 6 Algorithm A&AE&E enumerates (explicitly or implicitly) all colorings which are not worse than coloring ϕ_1. Since the number $f(G, q_1)$ of such colorings is finite, after a finite number of iterations of Algorithm A&AE&E, an optimal coloring ϕ^* has to be constructed. ■

Obviously, Algorithm A&AE&E is suitable for a mixed graph only with a small chromatic number (however, due to Remark 2, it is practically acceptable for problem $\mathcal{P}$) and when the first coloring ϕ_1 provides a good approximation to an optimal coloring, i.e. when the difference $q_1 - \gamma(G)$ is rather small.

Note that a similar idea of the enumeration of feasible colorings by increasing the current color of some vertex, for which there exists a chain to the vertex with the largest color, was previously used for a classical coloring of graph $(V, \emptyset, E)$ and provided good computational results (see e.g. [1]).

It is easy to show that the construction of an optimal coloring ϕ^* by Algorithm A&AE&E takes $O(n^2 f(G, q_1^n))$ time and $O(n^2)$ space. Indeed, at each iteration Algorithm A&AE&E has to store the input data, the lists P, S, $W(i)$ and $N(i)$, and at most two colorings, namely: the best coloring ϕ^* and the current coloring ϕ_r. All these sets use $O(n^2)$ space. Obviously, to realize Steps 1 - 3 takes $O(n^2)$ time. The number of colorings considered is no more than $O(f(G, q_1))$. The consideration of each coloring takes $O(n^2)$ time. Therefore, Algorithm A&AE&E takes $O(n^2 f(G, q_1))$ time. The function $f(G, t)$ will be studied in Section 6.

Algorithm A&AE&E may be used for coloring the mixed graph $G = (V, A, E)$ in different variants depending on the quality of the coloring which we need (see Remark 1). The first variant (we call it Algorithm A&AE) consists of Steps 1 - 4 and colors vertices of mixed graph G greedily in $O(n^2)$ time in accordance with vertex numbering $V = \{v^{(1)}, v^{(2)}, \dots, v^{(n)}\}$. In spite of the simplicity, Algorithm A&AE provides an optimal coloring for the mixed graph $G = (V, A, E)$ which has the following

Property 2: Inclusion $[v_i, v_j] \in E$ implies inclusion $(v_i, v_j) \in A$ or inclusion $(v_j, v_i) \in A$.

If the mixed graph G has not Property 2, then Algorithm A&AE constructs a coloring ϕ_1 without the guarantee that this coloring is optimal.

Along with the heuristic greedy Algorithm A&AE and the exact Algorithm A&AE&E, we consider the heuristic Algorithm AE&E which makes a partial enumeration of the colorings that may be better than ϕ_1. In the formal statement of Algorithm AE&E, Steps 1 - 5 are the same as in Algorithm A&AE&E, while Step 6 is as follows.

Step 6. **Selection of a vertex from set S_h which permits to increase its current color.**

6a) Set $C = S_h$.

6b) If $C = \emptyset$, then **STOP.**
Otherwise select vertex $v^{(t)} \in C$ which has the maximal superscript t among all vertices in the set C.

6c) Determine the smallest possible color $s \geq \phi_r(v^{(t)}) + 1$ which is feasible for vertex $v^{(t)}$ taking into account the colors $\phi_r(v^{(i)})$ of the vertices $v^{(i)} \in S_t$ (i.e. if $v^{(i)} \in S_t$, then $s \neq \phi_r(v^{(i)})$).

6d) If $s \geq q$, then set $C := C \setminus \{v^{(t)}\}$. If $C = \emptyset$, then set $C = S_t$. Go to Step 6b.
If $s < q$, then set $r := r + 1$, $d = t$, $l = s$ and go to Step 4.

Similarly to Algorithm A&AE, Algorithm AE&E constructs a coloring ϕ_r without the guarantee that this coloring is optimal (if the mixed graph G has not Property 1). We only know that coloring ϕ_r constructed by Algorithm AE&E uses no more colors than coloring ϕ_1 constructed by Algorithm A&AE.

5 Computational results

Algorithm A&AE&E, Algorithm A&AE and Algorithm AE&E were coded in FORTRAN and tested on a PC 486 (120 MHz) for coloring the vertices of randomly generated mixed graphs. Similarly to [3], to generate a mixed graph with $|A|$ arcs and $|E|$ edges we choose uniformly distributed $|A| + |E|$ integers $\tau_l \in \{1, 2, \dots, k\}$, where $k = \binom{n}{2}$, and τ_l denotes the pair of vertices with the ordinal number l of vertex indices: $\tau_1 = (v_1, v_2), \tau_2 = (v_1, v_3), \dots, \tau_k = (v_{n-1}, v_n)$. For each above $\tau_j = (v_r, v_s)$, it is assumed that $r < s$. This defines a labeled mixed graph of order n, with the arc set $A = \{\tau_1, \tau_2, \dots, \tau_{|A|}\}$ and with the edge set $E = \{\tau_{|A|+1}, \tau_{|A|+2}, \dots, \tau_{|A|+|E|}\}$.

In Tables 1, 2, 3, 4 and 5, computational results are given for randomly generated mixed graphs of orders 30, 40, 50, 70 and 100, respectively. For each considered combination of arc density and edge density, the programs colored the vertices of 10 randomly generated mixed graphs. In each table, column 5 presents the average number $UB1\ \gamma(G)$ of colors used in the first coloring ϕ_1 constructed, i.e. $UB1\ \gamma(G)$ denotes the first upper bound q_1 of the chromatic number $\gamma(G)$. Column 8 presents the average CPU time (in seconds) for constructing coloring ϕ_1. Column 6 presents the average number $UB2\ \gamma(G)$ of colors used in the best coloring ϕ_r constructed by Algorithm AE&E. Column 9 presents the average CPU time for constructing coloring ϕ_r. Column 10 presents the average CPU time for the whole realization of Algorithm AE&E. Column 7 presents the average chromatic number $\gamma(G)$. Column 11 presents the average CPU time for constructing an optimal coloring by

Table 1: Graphs of order 30

$\lvert A\rvert$	Arc density	$\lvert E\rvert$	Edge density	$UB1$ $\gamma(G)$	$UB2$ $\gamma(G)$	$\gamma(G)$	CPU A&AE ($UB1$)	CPU $UB2$	CPU AE&E	CPU $\gamma(G)$	CPU A&AE&E
1	2	3	4	5	6	7	8	9	10	11	12
0	0.0	43	0.1	3.9	3.0	3.0	0.00	0.09	0.17	0.10	0.18
43	0.1	43	0.1	4.5	4.1	3.9	0.01	0.04	0.14	0.79	1.27
87	0.2	43	0.1	5.5	5.1	4.7	0.01	0.05	0.14	0.30	1.51
130	0.3	43	0.1	6.3	6.0	5.5	0.02	0.04	0.14	1.23	48.14
174	0.4	43	0.1	6.3	6.3	6.2	0.02	0.02	0.12	0.03	5.71
217	0.5	43	0.1	7.6	7.6	7.5	0.02	0.02	0.14	0.03	44.24
261	0.6	43	0.1	7.4	7.4	7.3	0.03	0.03	0.15	0.03	1.97
304	0.7	43	0.1	8.0	8.0	8.0	0.03	0.03	0.13	0.04	17.72
348	0.8	43	0.1	8.4	8.4	8.4	0.03	0.03	0.14	0.04	1.87
391	0.9	43	0.1	8.3	8.3	8.3	0.03	0.03	0.14	0.04	5.13
435	1.0	43	0.1	7.9	7.9	7.9	0.05	0.05	0.16	0.05	1.27
0	0.0	87	0.2	5.0	4.1	4.0	0.00	0.12	0.22	1.53	1.69
43	0.1	87	0.2	6.0	5.5	5.0	0.00	0.05	0.17	2.10	14.54
87	0.2	87	0.2	7.6	7.4	6.5	0.00	0.03	0.96	38.12	597.80
130	0.3	87	0.2	8.6	8.4	7.3	0.02	0.04	0.26	76.81	328.62
174	0.4	87	0.2	9.5	9.2	8.6	0.02	0.05	0.18	180.53	445.11
217	0.5	87	0.2	9.7	9.5	9.3	0.02	0.05	0.24	0.53	67.52
261	0.6	87	0.2	10.0	9.9	9.7	0.02	0.03	0.16	1.26	21.10
304	0.7	87	0.2	11.0	10.8	10.5	0.04	0.06	0.20	1.48	15.16
348	0.8	87	0.2	11.8	11.7	11.6	0.05	0.06	0.18	1.16	116.74
391	0.9	87	0.2	10.5	10.5	10.5	0.05	0.05	0.18	0.05	16.22
435	1.0	87	0.2	11.6	11.6	11.6	0.04	0.04	0.19	0.05	28.85

Algorithm A&AE&E. Column 12 presents the average CPU time for the whole Algorithm A&AE&E, i.e. this time includes the CPU time for constructing an optimal coloring and for proving that coloring ϕ^* is optimal.

For the mixed graphs of orders 50, 70 and 100, only results for Algorithms A&AE and AE&E are presented in Tables 3, 4 and 5, however the ordinal numbers of the columns used are the same as in Tables 1 and 2.

As follows from the computational results, Algorithm A&AE&E proved optimality of the coloring ϕ^* for mixed graphs of an order of 40. For mixed graphs of an order larger than 40 (see Tables 3, 4 and 5), restricted versions of Algorithm A&AE&E (i.e. Algorithm A&AE and Algorithm AE&E) may be executed within a reasonable running time. The most critical parameter of Algorithm A&AE&E is the chromatic number $\gamma(G)$ and for the execution of Algorithm A&AE&E during a reasonable time, $\gamma(G)$ must be not much more than 10. Note however, that this restriction is not very important for the practical use in problem $\mathcal{P}$ (see Remark 2 and the computational results in [8]). The largest CPU time equaled to 36293.23 sec was used by Algorithm A&AE&E for one instance $G = (V, A, E)$ from the series of

Table 2: Graphs of order 40

$\lvert A \rvert$	Arc density	$\lvert E \rvert$	Edge density	$UB1$ $\gamma(G)$	$UB2$ $\gamma(G)$	$\gamma(G)$	CPU A&AE ($UB1$)	CPU $UB2$	CPU AE&E	CPU $\gamma(G)$	CPU A&AE&E
1	2	3	4	5	6	7	8	9	10	11	12
0	0.0	78	0.1	4.3	3.4	3.3	0.00	0.09	0.18	0.50	1.75
78	0.1	78	0.1	5.7	5.5	4.9	0.01	0.03	0.14	350.05	820.39
156	0.2	78	0.1	7.0	6.3	5.9	0.02	0.08	0.19	14.16	876.64
234	0.3	78	0.1	7.7	7.7	6.9	0.04	0.04	0.16	2.14	1000.68
312	0.4	78	0.1	8.7	8.5	7.8	0.05	0.07	0.18	2.11	413.55
390	0.5	78	0.1	9.4	9.3	8.9	0.07	0.08	0.17	1.04	1096.64
468	0.6	78	0.1	9.6	9.6	9.4	0.07	0.07	0.18	0.09	2332.64
545	0.7	78	0.1	9.7	9.7	9.7	0.10	0.10	0.20	0.10	761.11
624	0.8	78	0.1	9.9	9.9	9.8	0.10	0.10	0.23	0.11	135.33
701	0.9	78	0.1	10.8	10.8	10.7	0.11	0.11	0.23	0.13	328.48
780	1.0	78	0.1	10.9	10.9	10.9	0.12	0.12	0.24	0.12	229.25
0	0.0	156	0.2	6.0	5.0	4.8	0.01	0.11	0.52	1.14	3307.01
78	0.1	156	0.2	7.7	6.7	6.1	0.02	0.26	0.63	203.89	1866.61
156	0.2	156	0.2	10.0	9.1	7.9	0.04	0.23	1.43	1561.78	3859.24
234	0.3	156	0.2	11.0	10.3	9.2	0.04	0.13	0.67	1349.56	5957.25
312	0.4	156	0.2	11.7	11.2	10.3	0.05	0.14	0.87	531.09	2882.00
390	0.5	156	0.2	11.8	11.6	11.1	0.06	0.08	0.42	293.97	6355.31
468	0.6	156	0.2	13.1	12.7	12.1	0.07	0.11	0.34	31.01	2095.42
545	0.7	156	0.2	12.9	12.8	12.6	0.09	0.10	0.74	89.40	1669.68
624	0.8	156	0.2	13.9	13.9	13.9	0.10	0.10	0.42	0.11	1144.25
701	0.9	156	0.2	15.2	15.1	14.9	0.10	0.12	0.50	0.23	4870.43
780	1.0	156	0.2	15.3	15.3	15.3	0.13	0.13	0.50	0.13	1742.34

Table 3: Graphs of order 50

$\|A\|$	Arc density	$\|E\|$	Edge density	$UB1$ $\gamma(G)$	$UB2$ $\gamma(G)$	CPU A&AE ($UB1$)	CPU $UB2$	CPU AE&E
1	2	3	4	5	6	8	9	10
0	0.0	122	0.1	5.0	4.1	0.00	0.00	0.01
122	0.1	122	0.1	7.2	6.5	0.02	0.04	0.19
245	0.2	122	0.1	8.7	8.3	0.05	0.05	0.17
367	0.3	122	0.1	10.3	9.7	0.08	0.08	0.14
490	0.4	122	0.1	10.9	10.7	0.08	0.09	0.26
612	0.5	122	0.1	11.1	11.0	0.12	0.12	0.20
735	0.6	122	0.1	11.7	11.5	0.14	0.15	0.23
857	0.7	122	0.1	12.7	12.4	0.16	0.16	0.26
980	0.8	122	0.1	12.7	12.6	0.20	0.20	0.35
1102	0.9	122	0.1	13.1	13.1	0.23	0.23	0.39
1225	1.0	122	0.1	13.1	13.1	0.25	0.25	0.39
0	0.0	245	0.2	7.6	5.1	0.00	9.12	9.66
122	0.1	245	0.2	10.2	8.9	0.03	22.68	40.53
245	0.2	245	0.2	11.8	11.1	0.05	0.43	38.22
367	0.3	245	0.2	13.3	12.7	0.07	0.11	7.20
490	0.4	245	0.2	14.6	14.2	0.10	0.40	3.53
612	0.5	245	0.2	15.1	14.9	0.12	0.15	6.91
735	0.6	245	0.2	16.3	16.2	0.14	0.14	3.09
857	0.7	245	0.2	15.9	15.9	0.19	0.19	7.62
980	0.8	245	0.2	18.0	17.9	0.20	0.22	11.18
1102	0.9	245	0.2	18.4	18.4	0.23	0.23	1.33
1225	1.0	245	0.2	17.7	17.7	0.25	0.25	1.92

Table 4: Graphs of order 70

$\|A\|$	Arc density	$\|E\|$	Edge density	$UB1$ $\gamma(G)$	$UB2$ $\gamma(G)$	CPU A&AE ($UB1$)	CPU $UB2$	CPU AE&E
1	2	3	4	5	6	8	9	10
0	0.0	241	0.1	6.1	4.7	0.01	1.19	2.13
241	0.1	241	0.1	8.9	8.2	0.06	0.26	7.39
483	0.2	241	0.1	11.8	11.5	0.13	0.36	4.69
724	0.3	241	0.1	14.0	13.4	0.21	0.60	2.89
966	0.4	241	0.1	15.4	15.3	0.28	0.28	1.93
1207	0.5	241	0.1	15.2	14.9	0.37	0.37	0.94
1449	0.6	241	0.1	15.6	15.5	0.43	0.43	3.00
1690	0.7	241	0.1	17.2	17.2	0.50	0.50	1.96
1932	0.8	241	0.1	17.1	17.0	0.58	0.77	3.62
2173	0.9	241	0.1	17.9	17.9	0.66	0.66	1.53
2415	1.0	241	0.1	19.4	19.4	0.73	0.73	2.43
0	0.0	483	0.2	8.5	6.5	0.00	895.22	1220.22
241	0.1	483	0.2	13.4	11.3	0.06	604.81	4744.92
483	0.2	483	0.2	16.4	15.0	0.13	57.46	3684.35
724	0.3	483	0.2	17.4	16.3	0.20	6.45	707.87
966	0.4	483	0.2	20.9	20.2	0.28	20.75	674.52
1207	0.5	483	0.2	22.3	21.4	0.36	4.60	428.83
1449	0.6	483	0.2	21.9	21.6	0.41	0.48	583.87
1690	0.7	483	0.2	24.0	23.8	0.49	4.30	607.57
1932	0.8	483	0.2	23.5	23.3	0.56	10.29	484.66
2173	0.9	483	0.2	23.8	23.8	0.64	0.64	312.99
2415	1.0	483	0.2	26.1	26.1	0.72	0.72	89.09

Table 5: Graphs of order 100

$\|A\|$	Arc density	$\|E\|$	Edge density	$UB1$ $\gamma(G)$	$UB2$ $\gamma(G)$	CPU A&AE ($UB1$)	CPU $UB2$	CPU AE&E
1	2	3	4	5	6	8	9	10
0	0.0	495	0.1	7.0	5.6	0.00	32.15	150.76
495	0.1	495	0.1	13.0	11.8	0.20	33.18	264.54
990	0.2	495	0.1	15.6	14.9	0.40	0.71	81.83
1485	0.3	495	0.1	18.4	18.0	0.68	1.13	107.54
1980	0.4	495	0.1	20.3	20.1	0.91	3.28	324.02
2475	0.5	495	0.1	22.1	22.1	1.13	1.13	76.11
2970	0.6	495	0.1	23.5	23.4	1.31	1.31	34.83
3464	0.7	495	0.1	24.0	24.0	1.56	1.56	138.64
3960	0.8	495	0.1	25.7	25.7	1.78	1.78	162.98
4454	0.9	495	0.1	26.4	26.4	2.02	2.02	484.42
4950	1.0	495	0.1	26.7	26.7	2.31	2.31	54.57

instances with $n = 40$, $|A| = 390$ and $|E| = 156$. It is worth to note that the construction of an optimal coloring with $\gamma(G) = 13$ for this instance takes only 2.03 sec and so the rest of the above huge time was used to prove that this coloring is indeed optimal.

For a mixed graph with a larger chromatic number, it is possible to calculate $\gamma(G)$ without proving of its optimality or at least to calculate an upper bound of $\gamma(G)$ which is rather close to the exact value of $\gamma(G)$. Taking into account Remark 1, the latter property of Algorithm A&AE&E is also acceptable for problem $\mathcal{P}$.

6 The chromatic polynomial

As it was shown in Section 4, to find a coloring of the mixed graph $G = (V, A, E)$, it is sufficient to construct an extension $G_k^E = (V, A \cup A_k^E, E)$ of the mixed graph G, whose best coloring ϕ may be constructed by CPM:

$$\phi(v_i) = s_i(V, A \cup A_k^E, E) \text{ and } \gamma(G_k^E) = \max\{s_i(V, A \cup A_k^E, E) \ : \ v_i \in V\},$$

since any mixed graph G_k^E has Property 1 (see Section 4). An optimal coloring of an arbitrary mixed graph (if any) may be defined due to the best extension G_k^E, i.e. due to the extension which has the minimal possible critical weight:

$$\gamma(G_k^E) = \min\{\gamma(G_1^E), \gamma(G_2^E), \ldots, \gamma(G_\lambda^E)\}.$$

In this section, we study function $f(G, t)$ which may be used for calculating the number of feasible schedules. Remind also that function $f(G, t)$ is used in the asymptotic bound of the running time of Algorithm AE&E (see Section 4). It will be shown that, similarly to a usual coloring of a graph $(V, \emptyset, E)$ (see e.g. [4, 11]), to find all colorings ϕ of the mixed graph $G = (V, A, E)$ with colors from the set $\{1, 2, \ldots, t\}$, it is sufficient to transform the mixed graph G into a set of mixed graphs (V', A', E') which has the following

Property 3: Graph $(V', \emptyset, E')$ is complete.

Obviously, if the mixed graph has Property 3, it has Property 1, too. While to construct an extension G_k^E of the mixed graph G we added arcs to G (see Section 4), to calculate $f(G, t)$ we shall use adding edges and transformations $G|v_iv_j$, described in Section 2. In particular, the influence of adding an edge and identifying a pair of vertices on the value $f(G, t)$ is described by the following claim.

Lemma 3 *If v_i and v_j are non-adjacent vertices in graph $(V, \emptyset, E)$, then*

$$f(G, t) = f((V, A, E \cup \{[v_i, v_j]\}), t) + f(G|v_iv_j, t).$$

Proof: It is easy to see that the number of different colorings ϕ of the mixed graph G with colors from the set $\{1, 2, \ldots, t\}$ for which $\phi(v_i) \neq \phi(v_j)$, is equal to $f((V, A, E \cup \{[v_i, v_j]\}), t)$, while the number of different colorings ϕ of the mixed graph G for which $\phi(v_i) = \phi(v_j)$, is equal to $f(G|v_iv_j, t)$, and only these two cases are possible. ■

By applying Lemma 3 a finite number of times, we can obtain formulas (7) and (8) and Lemma 4 which follow.

If $H \subseteq E$ and graph $(V, \emptyset, E)$ is acyclic, then

$$f(G,t) = \sum (-1)^{n-|V|N|} f((V, A, E \setminus H)|N, t), \tag{7}$$

where the sum is taken over all subsets $N \subseteq H$ for which in digraph $(V, N, \emptyset)$, there does not exist a path connecting adjacent vertices in graph $(V, \emptyset, E \setminus H)$.

When representing graph $(V, \emptyset, E)$ as the union of acyclic graphs, due to (7) the problem of determining $f(G,t)$ can be reduced to an analogous problem for digraphs which was considered e.g. in [9].

Let the elements of set E be prescribed by the indices $1, 2, \ldots, |E|$, and the notation $[v_r, v_s]_k$ means that edge $[v_r, v_s] \in E$ has the index k, then the following equality holds:

$$f(G,t) = f((V, A, \emptyset), t) - \sum_{[v_i, v_j]_l \in E} f((V, A, E \setminus \cup_{k=1}^{l-1} \{[v_p, v_q]_k\})|v_i v_j, t). \tag{8}$$

The formula (8) allows to reduce the calculation of the value $f(G,t)$ for a mixed graph of order n to the calculation of these values for a mixed graph of order $n-1$ and the value $f((V, A, \emptyset), t)$, which was studied in [9].

As a label of each vertex $v \in V|N$ of the mixed graph $G|N$, we use a non-ordered set of indices of vertices of the mixed graph G which have been identified into vertex v.

Lemma 4 *If H is an arbitrary set of edges added to the graph $(V, \emptyset, E)$, then*

$$f(G,t) = \sum f((V, A, E \cup H)|N, t),$$

where the sum is taken over all subsets $N \subseteq H$ for which the labeled mixed graphs $G|N$ are different and in the digraph $(V, N, \emptyset)$, there does not exist a path connecting adjacent vertices in the graph $(V, \emptyset, E)$.

For the calculation of the value $f(G,t)$ by means of the known numbers of colorings of all extensions: $G_1^E, G_2^E, \ldots, G_\lambda^E)$, we can use the following claim.

Lemma 5 *If digraph $(V, A, \emptyset)$ has no circuits and graph $(V, \emptyset, E)$ has no loops, then*

$$f(G,t) = f(G_1^E, t) + f(G_2^E, t) + \ldots + f(G_\lambda^E, t).$$

Proof: For each edge $[v_i, v_j] \in E$ such that v_i and v_j are not adjacent in digraph $(V, A, \emptyset)$, we obtain the equality

$$f(G,t) = f((V, A \cup \{(v_i, v_j)\}, E), t) + f((V, A \cup \{(v_j, v_i)\}, E), t)$$

immediately due to Definition 1. Since digraph $(V, A, \emptyset)$ does not contain circuits, based on the above equality applied to each such edge $[v_i, v_j]$, one gets the validity of Lemma 5 by applying Lemma 1 and Theorem 2. ■

The algorithms of constructing all extensions of an arbitrary mixed graph have been described e.g. in [9] and in [10, pp. 294 - 297]. Lemmas 4 and 5 allow to calculate the value $f(G,t)$ by means of the calculation of value $f(G^{(*)},t)$, where the mixed graph $G^{(*)}$ has Property 1 and Property 2.

We denote by $\pi(V,A,\emptyset)$ the number of all linearly ordered sets $(v^{(1)},v^{(2)},\dots,v^{(n)})$ of the vertices of digraph $(V,A,\emptyset)$ such that each arc $(v^{(i)},v^{(j)})\in A$ implies inequality $i<j$. The calculation of the number $\pi(V,A,\emptyset)$ has been described e.g. in [9, 10].

Theorem 3 *If $E\cap H=\emptyset$ and graph $(V,\emptyset,E\cup H)$ is complete, then*

$$f(G,t)=\sum \pi((V,A,\emptyset)|N)\binom{t}{|V|N|},$$

where the sum is taken over all subsets $N\subseteq H$ for which the labeled mixed graphs $G|N$ are different and in digraph $(V,N,\emptyset)$, there does not exist a path connecting adjacent vertices in the graph $(V,\emptyset,E)$.

Proof: First, we consider a mixed graph G with the complete subgraph $(V,\emptyset,E)$, i.e. in this case we have $H=\emptyset$. If digraph $(V,A,\emptyset)$ contains a circuit, then $f(G,t)=0$ for any number $t\geq 1$ (see Theorem 2), and we have equality $\pi(V,A,\emptyset)=0$ (see Lemma 1).

Suppose now that digraph $(V,A,\emptyset)$ does not contain a circuit and that $G_k^E=(V,A\cup A_k^E,E)$ is an extension of the mixed graph G. Since graph $(V,\emptyset,E)$ is complete, the digraph $(V,A\cup A_k^E,\emptyset)$ is complete too, and for any coloring ϕ of the mixed graph G_k^E, inequalities of the form $\phi(v_{j_1})<\phi(v_{j_2})<\ldots<\phi(v_{j_n})$ hold. To satisfy these inequalities, there are exactly n different colors from the set $\{1,2,\dots,t\}$ required. And conversely, each of n different colors from the set $\{1,2,\dots,t\}$ define a coloring of the mixed graph G_k^E. Consequently, we have $f(G|E,t)=\binom{t}{n}$. It is easy to see that the number λ of different extensions of the mixed graph G with the complete subgraph $(V,\emptyset,E)$ is equal to $\pi(V,A,\emptyset)$.

Thus, if $(V,\emptyset,E)$ is the complete graph, then

$$f(G,t)=\pi(V,A,\emptyset)\binom{t}{n}. \tag{9}$$

In the case of an arbitrary mixed graph G without loops in subgraph $(V,\emptyset,E)$, Theorem 3 is also valid. Indeed, since graph $(V,\emptyset,E\cup H)$ is complete, the graphs $(V,\emptyset,E\cup H)|N,\ N\subseteq H$, are complete too, and the validity of Theorem 3 follows from Lemma 4. ■

Corollary 2 *If digraph $(V,A,\emptyset)$ has no circuits and graph $(V,\emptyset,E)$ has no loops, then $f(G,t)$ is a polynomial of degree n in the variable t and with the minimal degree of variable t with a non-zero coefficient being not smaller than the number of connected components in the mixed graph G.*

Proof: From Theorem 3 it follows that $f(G,t)$ is a polynomial in the variable t, where the highest degree is obtained for $N=\emptyset$.

Moreover, if $G_1, G_2, \ldots, G_k$ are the connected components of the mixed graph G, then obviously we have

$$f(G,t) = \Pi_{i=1}^{k} f(G_i, t),$$

and for the second part of Corollary 2, it is sufficient to take into account that the constant in the polynomial $f(G,t)$ is equal to 0 for any mixed graph G . ■

Corollary 3 *The sum of the coefficients of the polynomial $f(G,t)$ is equal to 0 if $E \neq \emptyset$, and it is equal to 1 if $E = \emptyset$. The coefficients a_n and a_{n-1} of t^n and t^{n-1} are defined as follows:*

$$a_n = \frac{\pi(V, A, \emptyset)}{n!}; \quad a_{n-1} = \frac{1}{(n-1)!}\Big[\sum_{[v_i,v_j]\notin E} \pi((V, A, \emptyset)|v_iv_j) + \frac{1-n}{2}\pi(V, A, \emptyset)\Big].$$

Proof: Obviously, the sum of the coefficients of the polynomial $f(G,t)$ is equal to $f(G,1)$. It is easy to see that $f(G,1) = 0$ if $E \neq \emptyset$, and $f(G,1) = 1$ otherwise.

The coefficients a_n and a_{n-1} can be determined immediately from Theorem 3. ■

Due to Corollary 3 it follows that the asymptotic bound of the running time of Algorithm A&AE&E (see Section 4) can be represented as $O(\frac{\pi(V,A,\emptyset)}{n!}n^2 q_1^n)$.

Returning to the scheduling example given in Section 3, we can calculate the chromatic polynomials for the mixed graph Γ_1^{12} presented in Figure 3b. Adding edge $[v_1, v_2]$ to the mixed graph Γ_1^{12} and identifying vertices v_1 and v_2 into one vertex v, we obtain the mixed graph $G_1 = (\{v_1, v_2, v_4\}, \{(v_2, v_1)\}, \{[v_1, v_2], [v_2, v_4]\})$ and the mixed graph $G_2 = (\{v, v_4\}, \{(v, v)\}, \{[v, v_4]\})$. Next, due to Lemma 3, we obtain $f(\Gamma_1^{12}, t) = f(G_1, t) + f(G_2, t)$. Then adding edge $[v_1, v_4]$ to the mixed graph G_1 and identifying vertices v_1 and v_4, we obtain the mixed graph $G_3 = (\{v_1, v_2, v_4\}, \{(v_2, v_1)\}, \{[v_1, v_2], [v_2, v_4], [v_1, v_4]\})$ and the mixed graph $G_4 = (\{v_2, v\}, \{(v_2, v)\}, \{[v_2, v]\})$. Due to Lemma 3, we obtain $f(G_1, t) = f(G_3, t) + f(G_4, t)$. Finally, due to formula (9) we obtain $f(\Gamma_1^{12}, t) = f(G_2, t) + f(G_3, t) + f(G_4, t) = 2\binom{t}{2} + 3\binom{t}{3} + \binom{t}{2} = 1/2t^3 - 1/2t$.

After a similar calculation for the mixed graph Γ_2^{12} (see Figure 3c), we obtain $f(\Gamma_2^{12}, t) = 1/6t^3 - 1/6t$.

7 Conclusion

As it was mentioned in Introduction, another definition of a mixed graph coloring ψ has been used in [3]:

Definition 2 *A function $\psi : V \to \{1, 2, \ldots, t\}$ is a coloring of the mixed graph G, if $\psi(v_i) < \psi(v_j)$ for each arc $(v_i, v_j) \in A$ and $\psi(v_p) \neq \psi(v_q)$ for each edge $[v_p, v_q] \in E$. The number of different functions ψ with values from the set $\{1, 2, \ldots, t\}$, which are such colorings of the mixed graph G, will be denoted by $f_\psi(G,t)$. The smallest number t for which $f_\psi(G,t) > 0$ will be denoted by $\gamma_\psi(G)$.*

Let $G = (V, A, E)$ be an arbitrary mixed graph. We denote by $(V, \emptyset, E \cup E_A)$ the graph obtained from G due to the replacement of each arc $(v_i, v_j) \in A$ by the edge $[v_i, v_j] \in E_A$. It is easy to prove the non-strict inequality

$$f_\psi(G, t) \leq f((V, \emptyset, E \cup E_A), t). \tag{10}$$

Indeed, each coloring ψ of the graph $(V, \emptyset, E \cup E_A)$ defines a unique orientation of each edge $[v_i, v_j] \in E_A$ as follows: if $\phi(v_i) < \phi(v_j)$, then we obtain the orientation (v_i, v_j), if $\phi(v_i) > \phi(v_j)$, then we obtain the orientation (v_j, v_i). Coloring ϕ will be a coloring ψ of the mixed graph G if and only if the above orientations of all edges E_A will be contained in the set A. Thus, there exists a surjective mapping of a part of the colorings ϕ of the graph $(V, A, E \cup E_A)$ into all colorings ψ of the mixed graph G, and inequality (10) holds.

From Definitions 1 and 2, it follows that ϕ is more general than ψ. Indeed, coloring ϕ turns into coloring ψ for each mixed graph $G = (V, A, E)$ which has Property 1. On the other hand, not each coloring ϕ can be represented as a coloring ψ if in the mixed graph G there exist arcs $(v_i, v_j) \in A$ with $[v_i, v_j] \notin E$.

Furthermore, we have the obvious non-strict inequality

$$f_\psi(G, t) \leq f(G, t), \tag{11}$$

which is realized as equality for $t \geq \gamma_\psi(G)$ if and only if the mixed graph G has Property 1.

Note also that in contrast to (10), the sign between the values $f(G, t)$ and $f((V, \emptyset, E \cup E_A), t)$ may be $>, =$ or $<$. As a conclusion from (10), (11) and the last remark, it follows that optimization and mostly enumeration questions for a mixed graph coloring ϕ are, in general, more difficult than those for a mixed graph coloring ψ. At least, an enumerative method for calculating $\gamma(G)$ requires to evaluate more colorings than an analogous method for calculating $\gamma_\psi(G)$.

In [3], a branch-and-bound algorithm (we call it Algorithm B&B) to find an optimal coloring ψ has been developed and tested. In every stage, Algorithm B&B constructs a partial coloring ψ of a mixed graph (V, A, E). The branching is done according to the color assigned to the current vertex v provided that some colors are forbidden due to colors previously assigned to vertices that are adjacent to v in graph $(V, \emptyset, E)$ and to vertices which precede v in digraph $(V, A, \emptyset)$. The current vertex v to be colored is one with the least number of forbidden colors. Note that Algorithm B&B has partial colorings during the coloring process and constructs a complete optimal coloring only at the end of the calculation. Of course, Algorithm B&B may be adopted for a more general coloring ϕ, too.

In this paper, we developed algorithms for calculating an optimal or a near optimal coloring ϕ which is suitable for scheduling problem $\mathcal{P}$. Algorithm A&AE&E (and Algorithms A&AE and AE&E) constructs a heuristic solution in $O(n^2)$ time which provides a rather good upper bound for $\gamma(G)$. This is important for the scheduling problem $\mathcal{P}$ since, if this bound will be smaller than or equal to m_k (see condition 2 in Theorem 1), then Algorithm A&AE&E may be stopped after $O(n^2)$ time. In the general case, the application of algorithm A&AE&E may be stopped when the best coloring constructed by Algorithm A&AE&E needs no more than m_k colors. As follows from the computational results, Algorithm A&AE&E is suitable for problem $\mathcal{P}$ since, if it is shown that the mixed graph $\Gamma_k[\alpha]$ is m_k-colorable, then Algorithm A&AE&E may be stopped. As follows from Tables 1 - 5, good colorings and even optimal ones were constructed rather quickly, and the most part of the CPU time was used to prove that the current coloring is optimal. However, to prove that the last coloring ϕ_{r^*} is optimal is only required in the case when $\gamma(\Gamma_k[\alpha]) > m_k$. The value m_k may be used as input parameter of Algorithm A&AE&E. The largest part of the computational time used to prove that the last best coloring is optimal, is not necessary for the most implementations of Algorithm A&AE&E for solving problem $\mathcal{P}$.

Note that algorithm A&AE&E calculates an optimal coloring ϕ within reasonable CPU time when $\gamma(G)$ is rather small which is also practically acceptable for problem $\mathcal{P}$ (see Remark 2 in Section 2). However, the exact Algorithm A&AE&E is not satisfactory for the mixed graph coloring problem when the chromatic number $\gamma(G)$ may be large. For such a mixed graph, another approach or a combination of different approaches have to be investigated. E.g. simple lower and upper bounds (6) from Corollary 1 or some others may be used within branch-and-bound methods, which may be a topic for future research.

Different questions also arise within the consideration of the chromatic polynomial of a mixed graph which is more complicated than that for a graph and that for a digraph. In Section 6, the calculation of $f(G,t)$ was based on a transformation of the mixed graph $G = (V, A, E)$ into a set of mixed graphs which have Property 3 (see Theorem 3). Such an approach is only suitable if set E has a sufficiently large cardinality. On the other hand, if $|E|$ is rather small, a transformation of the mixed graph $G = (V, A, E)$ into a set of digraphs may be more efficient.

Acknowledgment

The authors are indebted to G. V. Andreev for his qualified participation in the computational experiments.

References

[1] Christofides, N., (1975), "Graph theory: algorithmic approach", Academic Press, New York.

[2] Jain, A.S., Meeran, S., (1998), "The job-shop: past, present and future", European J. Oper. Res. (to appear).

[3] Hansen, P., Kuplinsky, J., de Werra, D., (1997), "Mixed graph colorings", Math. Meth. Oper. Res., 45, 145 – 160.

[4] Harary, F., (1969), "Graph theory", Addison-Wesley, Reading, MA.

[5] Karp, R.M., 1972, "Reducibility among combinatorial problems", in: R.E. Miller, J.W. Thather, eds., Complexity of Computer Computations, Plenum Press, New York, 85 – 103.

[6] Krüger, K., Sotskov, Yu.N., Werner, F., (1998), "Heuristics for generalized shop scheduling problems based on decomposition", Int. J. Prod. Res. 36 (11), 3013 – 3033.

[7] Kubale, M., (1989), "Interval vertex-coloring of a graph with forbidden colors", Discrete Math., 74, 125 – 136.

[8] Sotskov, Yu.N., (1997), "Mixed multigraph approach to scheduling jobs on machines on different types", Optimization, 42, 245 – 280.

[9] Stanley, R.P., (1973), "Acyclic orientations of graphs", Discrete Math., 5 (2), 171 – 178.

[10] Tanaev, V.S., Sotskov, Yu.N., Strusevich, V.A., (1994), "Scheduling theory. Multi-stage systems", Kluwer Academic Publishers, Dordrecht.

[11] Thulasiraman, K., Swany, M.N.S., (1992), "Graphs: theory and algorithms", John Wiley & sons, inc., Toronto.