algorithms

Article

Mixed Graph Colouring as Scheduling a Partially Ordered Set of Interruptible Multi-Processor Tasks with Integer Due Dates

Evangelina I. Mihova [1] and Yuri N. Sotskov [2,*]

[1] Mathematical Institute, Faculty of Mathematics, Computer Science and Statistics, Ludwig-Maximilians-Universitat Munich, Geschwister-Scholl-Platz, 1, 80539 Munich, Germany; evangelinamihova@gmail.com

[2] United Institute of Informatics Problems, National Academy of Sciences, 6 Surganov Street, 220012 Minsk, Belarus

* Correspondence: sotskov48@mail.ru; Tel.: +375-17-249-61-20

Abstract: We investigate relationships between scheduling problems with the bottleneck objective functions (minimising makespan or maximal lateness) and problems of optimal colourings of the mixed graphs. The investigated scheduling problems have integer durations of the multi-processor tasks (operations), integer release dates and integer due dates of the given jobs. In the studied scheduling problems, it is required to find an optimal schedule for processing the partially ordered operations, given that operation interruptions are allowed and indicated subsets of the unit-time operations must be processed simultaneously. First, we show that the input data for any considered scheduling problem can be completely determined by the corresponding mixed graph. Second, we prove that solvable scheduling problems can be reduced to problems of finding optimal colourings of corresponding mixed graphs. Third, finding an optimal colouring of the mixed graph is equivalent to the considered scheduling problem determined by the same mixed graph. Finally, due to the proven equivalence of the considered optimisation problems, most of the results that were proven for the optimal colourings of mixed graphs generate similar results for considered scheduling problems, and vice versa.

Keywords: scheduling; maximal lateness; makespan; multi-processor task; mixed graph colouring

check for updates

Citation: Mihova, E.I.; Sotskov, Y.N. Mixed Graph Colouring as Scheduling a Partially Ordered Set of Interruptible Multi-Processor Tasks with Integer Due Dates. *Algorithms* **2024**, *17*, 299. https://doi.org/10.3390/a17070299

Academic Editor: Roberto Montemanni

Received: 16 May 2024
Revised: 20 June 2024
Accepted: 1 July 2024
Published: 6 July 2024

1. Introduction

For optimal production planning and scheduling, it is required to construct an optimal schedule for processing a set of tasks (jobs) using existing equipment (processors). Optimisation of a schedule is a key factor in improving efficiency, as it allows a scheduler to reduce production costs, the time of the execution of incoming customer orders and the timely supply of production processes with raw materials and other components that are required for the manufacture of final products. Practical production scheduling (planning) problems are diverse in terms of conditions, restrictions and the objectives to be achieved by optimal schedules. To solve a large number of scheduling problems, special models and algorithms have been developed that take into account the specific conditions and objectives of concrete production planning. An extension of the application areas of scheduling algorithms can be based on a generalisation of scheduling models for complicated multi-stage systems in order to present different problem sets and to develop universal methods for constructing optimal schedules on the basis of the generalised scheduling models.

It is well known that the construction of the makespan-optimal schedule for fulfilling operations of equal duration reduces the construction of the optimal colouring of vertices of the graph. If it is required to consider precedence constraints specified on the operation set and the impossibility of processing two or more tasks simultaneously on the same processor, then, to determine the makespan-optimal schedule for fulfilling the unit-time operations, one can use the optimal colouring of the mixed graph [1,2].

In this paper, we focus on scheduling problems with integer operation durations and job release dates to minimize the makespan $C_{\max}$ or maximal lateness $L_{\max}$ of the given jobs with integer due dates.

Let $G = (V, A, E)$ be a mixed graph with the set $V = \left\{v_1, v_2, \ldots, v_{|V|}\right\} \neq \varnothing$ of vertices, the set A of arcs and the set E of edges. Each arc determines an ordered pair of vertices from the set V. Each edge determines an unordered pair of vertices. The mixed graph G considered in this paper does not contain loops, multiple arcs and multiple edges. For the empty set of arcs, we obtain the graph $G = (V, \varnothing, E)$. For the empty set of edges, we obtain the directed graph $G = (V, A, \varnothing)$ (the digraph for short).

In [3], a colouring of the mixed graph was defined.

Definition 1 (**[3]**). *A function $c : V \to \{1, 2, \ldots, t\}$ is called a colouring $c(G)$ of a mixed graph $G = (V, A, E)$ if the inequalities $c(v_i) \leq c(v_j)$ hold for all arcs $(v_i, v_j) \in A$ and the relations $c(v_p) \neq c(v_q)$ hold for all edges $[v_p, v_q] \in E$. A colouring $c(G)$ is optimal if this colouring uses a minimal number $\chi(G)$ of colours $c(v_i) \in \{1, 2, \ldots, t\}$. The minimal number $\chi(G)$ is called a chromatic number of the mixed graph G.*

If the equality $A = \varnothing$ holds, then the colouring $c(G)$ is an ordinary colouring of the graph $G = (V, \varnothing, E)$. The mixed graph (V, A, E) with the non-empty set of arcs and the non-empty set of edges may not be colourable [3].

Theorem 1 (**[3]**). *A colouring $c(G)$ of vertices of mixed graph $G = (V, A, E)$ exists if and only if the directed graph $(V, A, \varnothing)$ has no circuit containing the adjacent vertices in the graph $(V, \varnothing, E)$.*

Finding an optimal colouring of the vertices of the mixed graph $G = (V, A, E)$ is an NP-hard problem, even if $A = \varnothing$; see [4]. The NP-hardness of the optimisation problem means that it is unlikely that there exists a polynomial algorithm for solving this problem, since there is a polynomial reduction to this problem from a huge number of NP-hard problems, while there is no polynomial algorithm for each of them yet.

In [1,2], it is proven that a job-shop problem with equal (unit) durations of operations with the objective of minimising a schedule length (i.e., makespan) reduces to a problem of finding the optimal colouring of the mixed graph $G = (V, A, E)$.

In this paper, we show that finding the optimal colouring $c(G)$ reduces to the problem of constructing a schedule that minimizes a maximal lateness for processing multi-processor tasks with integer durations, integer due dates and allowed operation interruptions. The converse is also proven. Processing a multi-processor task requires several dedicated processors; see Chapter 10 of the monograph [5]. Moreover, two types of precedence constraints can be specified for the multi-processor tasks, and the unit-time operations of the indicated subset must be processed simultaneously.

The remaining part of this paper is organised as follows. The settings of considered scheduling problems, preliminary results and notation are given in Section 2. Related papers are cited, where it has been shown that the problem of finding an optimal colouring of the mixed graph reduces to finding a makespan-optimal schedule for partially ordered operations with unit processing times. Scheduling problems are discussed in increasing order of their complexity. Section 3 demonstrates that the problem of optimal colouring of the mixed graph reduces to finding a makespan-optimal schedule of the multi-processor tasks with equal durations. In Section 4, we discuss scheduling problems of minimising maximal lateness of completing jobs with respect to integer due dates. New results are proven, demonstrated, and discussed in Sections 4–8. The possibility of interruptions processing integer-time tasks is studied in Section 5. A theorem is proven on the reduction of the optimal colouring $c(G)$ to the scheduling problems with a minimisation of the makespan for the interrupted integer–time tasks with integer release dates of the jobs. An illustrative example of the studied scheduling problem is solved in Section 6. Section 7 includes a theorem on the equivalence of scheduling problems with a maximal lateness objective and the problem of finding optimal colourings of the mixed graphs. Due to the proven equivalence of the problems of constructing optimal schedules and mixed graph colouring problems, many claims that have been proven for scheduling problems can be applied to the problems of optimal mixed graph colourings, and vice versa. In Section 8, we discuss new findings and their potential implications. Section 9 contains concluding remarks and promising future research directions. The new theorems and most of the theorems presented in the survey section are demonstrated in the numerical examples in the main text and in Appendices A and B.

2. Related Works, Definitions, and Preliminaries

In the following, terminology from [5] is used for scheduling problems and terminology from [6,7] is used for graph theory. To classify scheduling problems, a three-field notation $\alpha|\beta|\gamma$ [5] is used, where α determines the processing system, field β determines the characteristics of the jobs, and field γ an objective function.

2.1. Minimum Length Unit-Time Job-Shop Problems

We begin by examining the unit-time scheduling problem $J\big|t_{ij}=1\big|C_{\max}$ of constructing a makespan-optimal schedule for fulfilling the set $J=\left\{J_1,J_2,\dots,J_{|J|}\right\}$ of the given jobs by the dedicated processors $M=\left\{M_1,M_2,\dots,M_{|M|}\right\}$. The execution of the job $J_i\in J$ entails the execution of the ordered set $Q^{(i)}=\left\{Q_{i1},Q_{i2},\dots,Q_{i,|Q^{(i)}|}\right\}$ of operations with unit (equal) durations $t_{ij}=1$ in the strict linear order $(Q_{i1},Q_{i2},\dots,Q_{i,|Q^{(i)}|})$ on the corresponding processors from the set M. The operation $Q_{ij}\in Q^{(i)}$ is performed by the processor $M_{\mu(ij)}\in M$.

This processing system is called a job-shop, denoted $\alpha=J$ in the three-field notation $J\big|t_{ij}=1\big|C_{\max}$ used for this problem. The notation $C_{\max}$ is used to specify the optimality criterion of minimising the makespan $C_{\max}=\max\{C_i:J_i\in J\}$ (the schedule length) in the desired optimal schedule, where C_i denotes the completion time of the job $J_i\in J$. All jobs in the set $J=\left\{J_1,J_2,\dots,J_{|J|}\right\}$ are ready for processing from the same release date $r=0$. Interruptions are not allowed during the execution of an operation $Q_{ij}\in Q^{(i)}$ with $J_i\in J$. A schedule S for processing the job set $J=\left\{J_1,J_2,\dots,J_{|J|}\right\}$ is determined either by the start times $S(Q_{ij})\geq 0=t$ or by the completion times $C(Q_{ij})=S(Q_{ij})+t_{ij}$ of the operations $Q_{ij}\in Q:=\cup_{i=1}^{|J|}Q^{(i)}$. The length (makespan) of the schedule S is equal to $C_{\max}(S)=\max\left\{C(Q_{ij}):Q_{ij}\in Q\right\}$.

Let the subset Q_k of the set Q consist of operations to be fulfilled by the processor $M_k\in M$. Given that every pair of operations within the set Q_k must not be executed simultaneously, any schedule that exists for a job-shop problem $J\big|t_{ij}=1\big|C_{\max}$ determines $|M|$ strict linear orders of the operations Q_k fulfilled by the processors $M_k\in M$. An optimal schedule for the problem $J\big|t_{ij}=1\big|C_{\max}$.

$$S=(C(Q_{1,1}),C(Q_{1,2}),\dots,C(Q_{1,|Q^{(1)}|}),C(Q_{2,1}),C(Q_{2,2}),\dots,C(Q_{2,|Q^{(2)}|}),\dots, \qquad (1)$$
$$C(Q_{|J|,1}),C(Q_{|J|,2}),\dots,C(Q_{|J|,|Q^{(|J|)}|}))$$

has the minimum length $C_{\max}(S)$ among all schedules that exist for this problem.

The objective function for each regular criterion is a non-decreasing function of the arguments $C_i:=C(Q_{i,|Q^{(i)}|})$, where $J_i\in J$. When solving the scheduling problem $\alpha|\beta|\gamma$ with a regular criterion γ, the optimal schedule can be found within the set of semiactive schedules [5].

Definition 2 ([5]). *The schedule* **S** *for the problem* $\alpha|\beta|\gamma$ *is semiactive, if the execution of every operation cannot be processed earlier without violating the operation order in schedule* **S** *or another operation is processed later than in this schedule.*

2.2. Optimal Strict Colourings of Mixed Graphs and Equivalent Unit-Time Job-Shop Problems

The paper [3] introducing Definition 1 was published in Russian. In 1997, another colouring of mixed graph $c_<(G)$ was introduced in the paper [8], published in English.

Definition 3 ([8]). *The function* $c_< : V \to \{1, 2, \ldots, t\}$ *is a strict colouring* $c_<(G)$ *of the mixed graph* $G = (V, A, E)$*, if the inequalities* $c_<(v_i) < c_<(v_j)$ *hold for all arcs* $(v_i, v_j) \in A$ *and the relations* $c_<(v_p) \neq c_<(v_q)$ *hold for all edges* $[v_p, v_q] \in E$*. The strict colouring* $c_<(G)$ *is optimal, if this colouring uses a minimum number of colours* $c_<(v_i) \in \{1, 2, \ldots, t\}$*.*

We can use colouring $c(G)$ instead of strict colouring $c_<(G)$ of vertices of the mixed graph $G = (V, A, E)$ provided that the following implication holds for every arc $(v_i, v_j) \in A$:

$$(v_i, v_j) \in A \Rightarrow \ [v_i, v_j] \in E \tag{2}$$

If the mixed graph $G = (V, A, E)$ includes the arc $(v_i, v_j) \in A$ and the implication (2) does not hold, one can add an edge $[v_i, v_j]$ to the mixed graph G for each such arc $(v_i, v_j) \in A$. A strict colouring $c_<(G)$ of the mixed graph G can be represented as a colouring $c(G)$ of the mixed graph obtained by including all such edges to the mixed graph G. Thus, the following lemma is obtained.

Lemma 1. *A strict colouring* $c_<(G)$ *of the mixed graph* $G = (V, A, E)$ *can be interpreted as a special case of the colouring* $c(G)$*, if the inclusion* $(v_i, v_j) \in A$ *implies the inclusion* $[v_i, v_j] \in E$*.*

A criterion for the existence of a strict colouring of vertices of the mixed graph $G = (V, A, E)$ is determined by Corollary 1 of Theorem 1.

Corollary 1 ([1]). *A strict colouring* $c_<(G)$ *exists if and only if the digraph* $(V, A, \varnothing)$ *does not contain a circuit.*

Theorem 2 was proven in [1].

Theorem 2 ([1]). *The unit-time scheduling problem* $J|t_{ij} = 1|C_{\max}$ *is equivalent to the problem of finding the optimal strict colouring of vertices of the mixed graph* $G = (V, A, E)$ *with* $V = Q$*, provided that two conditions hold as follows:*

(a) $(Q, A, \varnothing) = \cup_{i=1}^{|J|}(Q_i, A_i, \varnothing)$*, where each subgraph* $(Q_i, A_i, \varnothing)$ *is a path passing through the vertices* Q_i *and equality* $Q_i \cap Q_j = \varnothing$ *holds for indexes* $i \neq j$*;*

(b) $(Q, \varnothing, E) = \cup_{k=1}^{|M|}(Q^{(k)}, \varnothing, E^{(k)})$*, where each subgraph* $(Q^{(k)}, \varnothing, E^{(k)})$ *is a complete graph on the set* Q_i *and equality* $Q^{(k)} \cap Q^{(l)} = \varnothing$ *holds for indexes* $k \neq l$*.*

Due to Lemma 1 and Theorem 2, one can obtain the following corollary.

Corollary 2 ([1]). *The unit-time scheduling problem* $J|t_{ij} = 1|C_{\max}$ *is equivalent to the problem of determining the optimal colouring of the mixed graph* $G = (V, A, E)$*, where the relations*

$$\left\{v_1, v_2, \ldots, v_{|V|}\right\} = V = Q = \left\{Q_{1,1}, Q_{1,2}, \ldots, Q_{1,|Q_1|}, Q_{2,1}, Q_{2,2}, \ldots, Q_{2,|Q_2|}, \ldots, Q_{|J|,1}, Q_{|J|,2}, \ldots, Q_{|J|,|Q_{|J|}|}\right\}$$

hold along with the above conditions (2), (a) and (b).

To illustrate Corollary 2, we consider Example 1 of the problem $J|t_{ij} = 1|C_{\max}$, whose input data are determined by the corresponding mixed graph G.

Remark 1. *Since all operations* $Q_{ij} \in Q_i$ *in the unit-time scheduling problem* $J|t_{ij} = 1|C_{\max}$ *have unit processing times, we represent each operation as a vertex in the mixed graph* $G = (V, A, E)$ *with indexing the vertices in the set* V *in the lexicographic order of the pairs* $\{i, j\}$ *of the indexes* *of the corresponding operations* $Q_{ij} \in Q_i$*,* $J_i \in J$*. Thus, we obtain the following set* $V = \left\{v_1, v_2, \ldots, v_{|V|}\right\}$ *of vertices, where the equality* $\left|V\right| = \sum_{i=1}^{|J|} \left|Q^{(i)}\right|$ *holds.*

Example 1. *The dedicated processors* $M = \{M_1, M_2, M_3, M_4\}$ *must process jobs* $J = \{J_1, J_2, J_3, J_4, J_5\}$*, where processing the job* $J_1 \in J$ *consists of the singleton* $V^{(1)} = \{v_1\}$*. The job* $J_2 \in J$ *is represented as a union of the path* (v_2, v_3) *in the digraph* $(V, A, \varnothing)$ *and the chain* (v_2, v_3) *in the graph* $(V, \varnothing, E)$*. The job* $J_3 \in J$ *contains the set* $V^{(3)} = \{v_4, v_5, v_6, v_7\}$ *of the ordered operations* (v_4, v_5, v_6, v_7)*. The job* $J_3 \in J$ *is represented as a union of the path* (v_4, v_5, v_6, v_7) *in the digraph* $(V, A, \varnothing)$ *and the chain* (v_4, v_5, v_6, v_7) *in the graph* $(V, \varnothing, E)$*. The job* $J_4 \in J$ *contains the set* $V^{(4)} = \{v_8, v_9, v_{10}\}$ *of the linearly ordered operations* (v_8, v_9, v_{10})*. The job* $J_4 \in J$ *is represented as a union of the path* (v_8, v_9, v_{10}) *in the directed graph* $(V, A, \varnothing)$ *and the* (v_8, v_9, v_{10}) *in graph* $(V, \varnothing, E)$*. The job* $J_5 \in J$ *contains the set* $V^{(5)} = \{v_{11}, v_{12}\}$ *of the ordered operations* (v_{11}, v_{12})*. The job* $J_5 \in J$ *is represented as a union of the path* (v_{11}, v_{12}) *in directed graph* $(V, A, \varnothing)$ *and the chain* (v_{11}, v_{12}) *in graph* $(V, \varnothing, E)$*.*

Let the set $V_1 = \{v_1, v_5, v_7\}$ be defined as a set of operations processed on the processor M_1. The forbiddance to process the operations of the set V_1 simultaneously is determined by the clique $\{v_1, v_5, v_7\}$ in the graph $(V, \varnothing, E)$; see Figure 1. Let the processor M_2 have to process the operations $V_2 = \{v_2, v_4\}$. The forbiddance to process operations of the set V_2 simultaneously is determined by the edge $[v_2, v_4]$. Let the processor M_3 have to process the operations $V_3 = \{v_3, v_9, v_{11}\}$. The forbiddance to process operations of the set V_3 simultaneously is determined by the clique $\{v_3, v_9, v_{11}\}$. Let the processor M_4 have to process the operations $V_3 = \{v_9, v_{11}, v_{15}\}$. The forbiddance to process any pair of operations of the set V_3 simultaneously is determined by the clique $\{v_9, v_{11}, v_{15}\}$. Let the processor M_4 have to process the operations $V_4 = \{v_6, v_8, v_{10}, v_{12}\}$. The forbiddance to process any pair of operations of the set V_4 simultaneously is determined by the clique $\{v_6, v_8, v_{10}, v_{12}\}$.

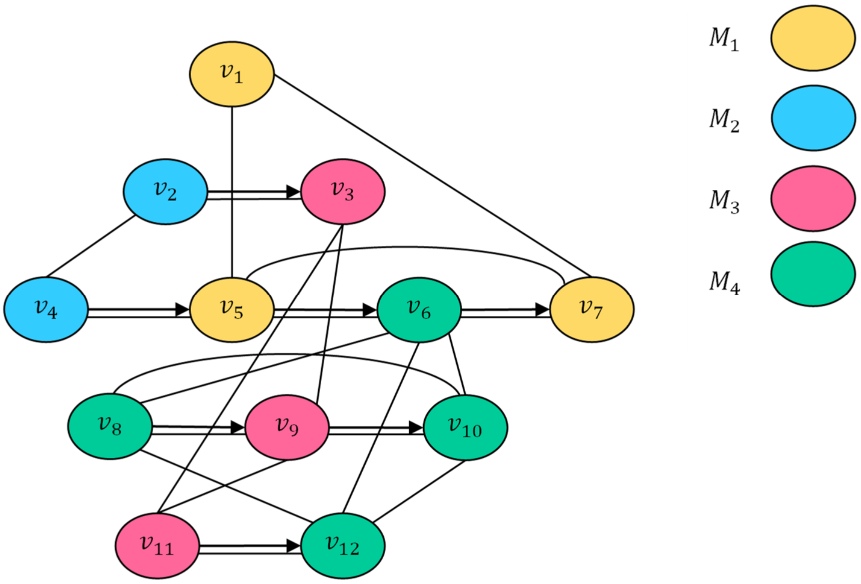

Figure 1. Mixed graph G that determines input data for Example 1.

A specific colour is used for indicating the operations $V_i = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\} \subseteq V$ that are processed on the processor $M_i \in M$; see Figure 1. It is clear that the mixed graph depicted in Figure 1 possesses properties (1), (a) and (b), and the indexing of all vertices of the set V satisfies Remark 1. The optimal colouring of the mixed graph G is defined as follows: $\mathrm{c}(\mathrm{v}_1) = 1$, $\mathrm{c}(\mathrm{v}_2) = 2$, $\mathrm{c}(\mathrm{v}_3) = 3$, $\mathrm{c}(\mathrm{v}_4) = 1$, $\mathrm{c}(\mathrm{v}_5) = 2$, $\mathrm{c}(\mathrm{v}_6) = 3$, $\mathrm{c}(\mathrm{v}_7) = 4$, $\mathrm{c}(\mathrm{v}_8) = 1$, $\mathrm{c}(\mathrm{v}_9) = 2$, $\mathrm{c}(\mathrm{v}_{10}) = 4$, $\mathrm{c}(\mathrm{v}_{11}) = 1$ and $\mathrm{c}(\mathrm{v}_{12}) = 2$. The optimality of the colouring c(G) follows from the fact that the graph $(V, \varnothing, E)$ contains the chain

(v_4, v_5, v_6, v_7) with four vertices and the digraph $(V, A, \varnothing)$ contains the path (v_4, v_5, v_6, v_7). So, the chromatic number $\chi(G)$ is not less than four. The optimality of the colouring c(G) follows also from the clique $\{v_6, v_8, v_{10}, v_{12}\}$.

Due to Corollary 2, an optimal colouring $c(G)$ defines the following optimal semiactive schedule S for Example 1 with the input data shown in Figure 1:

$$S = (C(Q_{1,1}) = 1, C(Q_{2,1}) = 2, (Q_{2,2}) = 3, C(Q_{3,1}) = 1, C(Q_{3,2}) = 2, C(Q_{3,3}) = 3, C(Q_{3,4}) = 4,$$
$$C(Q_{4,1}) = 1, C(Q_{4,2}) = 2, C(Q_{4,3}) = 4, C(Q_{5,1}) = 1, C(Q_{5,2}) = 2$$

Due to Corollary 2, we obtain the chromatic number $\chi(G) = 4 = C_{max}(S)$.

2.3. General Shop Minimum-Length Unit-Time Scheduling Problems

The general shop scheduling problem $G_c|t_{ij} = 1|C_{\max}$ is a generalisation of the job-shop scheduling problem $J|t_{ij} = 1|C_{\max}$. In the former problem $G_c|t_{ij} = 1|C_{\max}$, in addition to the strict order that is determined on the set of operations $Q^{(i)}$ of the same job $J_i \in J$, precedence constraints can be defined between operations of different jobs. Moreover, indicated unit-time operations must be processed simultaneously. In [9], completion-start precedence constraints and start-start precedence constraints between operations of different jobs were also considered.

The processing job $J_k \in J$ consists of the set $V^{(k)}$ of linearly ordered operations. The duration t_i of every operation v_i of the set $V = \cup_{k=1}^{|J|} V^{(k)}$ equals one; $t_i = 1$. We represent the job $J_k \in J$ as a union of the path $(v_{k_1}, v_{k_2}, \ldots, v_{k_{r_k}})$ in the directed graph $(V, A, \varnothing)$ and the chain $(v_{k_1}, v_{k_2}, \ldots, v_{k_{r_k}})$ in the graph $(V, \varnothing, E)$, which are subgraphs of the mixed graph $G = (V, A, E)$ determining input data of the problem $G_c|t_{ij} = 1|C_{\max}$. Thus, we obtain the vertex set $V = \cup_{k=1}^{|J|} V^{(k)}$ of the desired mixed graph $G = (V, A, E)$, the subset $E^* = \cup_{k=1}^{|J|} \left\{ [v_{k_1}, v_{k_2}], [v_{k_2}, v_{k_3}], \ldots, [v_{k_{r_k}-1}, v_{k_{r_k}}] \right\}$ of edges $E \supseteq E^*$ and the subset A^* of arcs A such that the implication (3) holds:

$$[v_i, v_j] \in E^* \Rightarrow\ (v_i, v_j) \in A^* \tag{3}$$

In the general shop unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$, along with a linear strict order given on the set $V^{(k)}$ of all operations of the job $J_k \in J$, there are given precedence constraints between operations belonging to different jobs in the set J Let $A \backslash A^*$ denote a subset of the set A such that the implication (2) does not hold for each arc $(v_i, v_j) \in A \backslash A^*$. In the unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$, one processor of the set M is needed to process an operation $v_i \in V = \cup_{k=1}^{|J|} V^{(k)}$. Let the set $V_i = \left\{ v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}} \right\} \subseteq V$ be a set of operations, which must be processed on the processor $M_i \in M$. A pair of operations processed on the processor $M_i \in M$ cannot be processed simultaneously. Such a capacity constraint for processing operations $V_i \subseteq V$ on the processor $M_i \in M$ is represented by the clique $\left\{ v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}} \right\}$ in the subgraph $(V, \varnothing, E \backslash E^*)$ of the desired mixed graph $G = (V, A, E)$.

The general shop unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$ is to find a schedule S for processing the set $V = \cup_{i=1}^{|M|} V_i = \cup_{k=1}^{|J|} V^{(k)}$ of partially ordered operations, whose length $C_{\max}(S)$ is minimal among the lengths of all semiactive schedules existing for the general shop scheduling problem $G_c|t_{ij} = 1|C_{\max}$. The input data for the general shop unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$ are determined by the mixed graph $G = (V, A, E)$, where the vertex set V is a set of operations, the arc set A determines precedence constraints and the edge set E determines capacity constraints.

The union $A^* \cup E^*$ of the arc set A^* and the edge set E^* determines $|J|$ subsets $V^{(k)}$ of the linearly ordered operations of the jobs $J_k \in J$. The subset $E \backslash E^*$ of the edges determines $|M|$ cliques $\left\{ v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}} \right\}$ in the subgraph $(V, \varnothing, E \backslash E^*)$, where all operations $\left\{ v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}} \right\}$ must be processed on the same processor $M_i \in M$. The precedence constraints between operations belonging to different jobs are determined by the directed subgraph $(V, A \backslash A^*, \varnothing)$ of the mixed graph $G = (V, A, E)$.

If the completion time c_{k_p} of the operation v_{k_p}, $J_k \in J$, must precede the start time s_{l_q} of the operation v_{l_q} belonging the job $J_l \in J$ with $k \neq l$, then the mixed graph $G = (V, A, E)$ must contain both arc $(v_{k_p}, v_{l_q}) \in A$ and edge $[v_{k_p}, v_{l_q}] \in E$; see Definition 1. We denote this completion-start constraint as $v_{k_p} \Rightarrow v_{l_q}$. If the constraint $v_p \Rightarrow v_q$ holds for the unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$, then the implication (1) holds for the arc $(v_p, v_q) \in A$ (the implication $(v_p, v_q) \in A \Rightarrow [v_p, v_q] \in E^*$ holds for the edges $E^* \subseteq E$). If the start time s_{k_p} of the operation v_{k_p} of the job $J_k \in J$ must precede the start time s_{l_q} of the operation v_{l_q}, $J_l \in J$, where $k \neq l$, then the desired mixed graph $G = (V, A, E)$ contains arc $(v_{k_p}, v_{l_q}) \in A$. We denote this weak precedence constraint as follows: $v_{k_p} \rightarrow v_{l_q}$. If the precedence constraint $v_p \rightarrow v_q$ holds for the problem $G_c|t_{ij} = 1|C_{\max}$, then the implication (1) does not hold for the arc $(v_p, v_q) \in A$.

Due to the use of weak precedence constraints, it is possible to require that a subset $V(k) \subset V$ of the unit-time operations must be processed simultaneously. To represent such a requirement, it is necessary to construct a circuit containing all vertices of the set $V(k)$. As shown in [9], the constructed mixed graph $G = (V, A, E)$ completely determines the input data for a general shop unit-time problem $G_c|t_{ij} = 1|C_{\max}$.

Definition 4. *If a mixed graph $G = (V, A, E)$ completely determines the input data for the scheduling problem $\alpha|\beta|\gamma$, then this individual problem $\alpha|\beta|\gamma$ is called a scheduling problem $\alpha|\beta|\gamma$ on the mixed graph G.*

Theorem 3 was proved in [9].

Theorem 3 ([9]). *A general shop unit-time problem $G_c|t_{ij} = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ reduces to the problem of determining the optimal colouring $c(G)$.*

It should be noted that there are problems in finding optimal colourings of the mixed graph that cannot be represented as a general shop unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$. In Appendix A, we solve Example A1 of the unit-time scheduling problem $G_c|t_{ij} = 1|C_{\max}$ to illustrate Theorem 3.

3. Colouring Vertices of Mixed Graphs and Unit-Time Scheduling Problems

In contrast to shop scheduling problems $J|t_{ij} = 1|C_{\max}$, $G_c|t_{ij} = 1|C_{\max}$ and other scheduling problems [5], where one machine performs each operation (task), in the multi-stage system, with multi-processor tasks [5,10,11], the task may require one processor or several dedicated processors during the entire time of processing this task. The tasks that require a common processor cannot be processed simultaneously.

Chapter 10 of the monograph [5] is devoted to unit-time scheduling problems $GMPT|t_{ij} = 1|C_{\max}$ with multi-processor tasks. In [9], the more general unit-time scheduling problem $G_cMPT|t_{ij} = 1|C_{\max}$ was introduced including the unit-time scheduling problem $GMPT|t_{ij} = 1|C_{\max}$. In the unit-time scheduling problem $G_cMPT|t_{ij} = 1|C_{\max}$, two types of the precedence relations $v_{k_p} \rightarrow v_{l_q}$ and $v_{k_h} \Rightarrow v_{l_g}$ can be given in the input data. It may also be required that a set $V(k) = \left\{v_{k_1}, v_{k_2}, \dots, v_{k_{|V(k)|}}\right\} \subseteq V$ of multi-processor tasks has to be performed simultaneously in each schedule. To present this condition, the circuit $(v_{k_1}, v_{k_2}, \dots, v_{k_{|V(k)|}}, v_{k_1})$ must be included in the digraph $(V, A, \varnothing)$ such that the following is true:

$$A'_c = \left\{(v_{k_1}, v_{k_2}), (v_{k_2}, v_{k_3}), \dots, (v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}), (v_{k_{|V(k)|}}, v_{k_1})\right\} \subseteq A.$$

We assume that the input data of the considered unit-time scheduling problem $G_cMPT|t_{ij} = 1|C_{\max}$ includes w subsets $V(1), V(2), \dots, V(w)$ of the given set

$V = \{v_1, v_2, \ldots, v_n\}$ such that all tasks of the subset $V(k)$ must be performed simultaneously; $k \in \{1, 2, \ldots, w\}$. Since all unit-time tasks of the set $V(k) = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\}$ must be fulfilled simultaneously, we obtain the following set of arcs:

$$A_c = \bigcup_{k=1}^{w} \left\{(v_{k_1}, v_{k_2}), (v_{k_2}, v_{k_3}), \ldots, (v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}), (v_{k_{|V(k)|}}, v_{k_1})\right\} \subseteq A \tag{4}$$

The example of the unit-time problem $G_cMPT|t_{ij} = 1|C_{\max}$ defines the mixed graph $G = (V, A, E)$ with arc set $A \supseteq A_c$. Theorems 4 and 5 are proven in [9].

Theorem 4 ([9]). *The unit-time problem $G_cMPT|t_{ij} = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ is solvable if and only if the directed subgraph $(V, A, \varnothing)$ has no circuit containing adjacent vertices in the subgraph $(V, \varnothing, E)$.*

Theorem 5 ([9]). *Every solvable unit-time problem $G_cMPT|t_{ij} = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ is equivalent to the problem of finding an optimal colouring $c(G)$ of the mixed graph G. For every colourable mixed graph $G = (V, A, E)$ there exists a unit-time problem $G_cMPT|t_{ij} = 1|C_{\max}$ determined by the same mixed graph G, which is equivalent to finding an optimal colouring $c(G)$ of the mixed graph G.*

Theorems 4 and 5 are illustrated by solving Example A2 of the problem $G_cMPT|t_{ij} = 1|C_{\max}$ on the mixed graph G depicted in Figure A2; see Appendix B.

4. Minimising Maximal Lateness and Equivalent Minimising Makespan for Jobs with Integer Release Dates

The unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ with integer release dates $r_i \geq 0$ of the jobs $J_i \in J$ has been studied in [12]. Since release dates r_i are non-negative integers and operations have unit durations, the input data of the problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ can be determined by the mixed graph G; see Sections 2 and 3.

4.1. Minimising Makespan for Unit-Time Tasks as Optimal Mixed Graph Colouring

In the problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$, the multi-processor task $v_i \in V$ is considered as a job J_i. A job consisting of a single operation is referred to as a simple job in Theorem 6 which follows. Note that the first part of this theorem has been proven in [12] by generating a mixed graph $G\prime$, which determines the problem $G_cMPT|t_{ij} = 1|C_{\max}$ to the mixed graph $G = (V, A, E)$ determining the input data for the problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$. Due to the demonstrated correspondence of the terms used in optimal coloring $c(G)$ and those used in solving a unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$, the following first part of Theorem 6 was proven in [12]: *Any solvable unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ is equivalent to the problem of finding an optimal colouring of the mixed graph $G = (V, A, E)$.*

Theorem 6. *Any solvable unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ on the mixed graph G is equivalent to the problem of finding an optimal colouring of the mixed graph G. Conversely, for any colourable mixed graph G, there exists a unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ on the mixed graph G, which is equivalent to the problem of finding optimal colouring $c(G)$.*

Proof of the second part of Theorem 6. We next prove the claim as follows: *for any colorable mixed graph $G = (V, A, E)$, there exists a unit-time problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ on the mixed graph G, which is equivalent to the problem of finding optimal colouring $c(G)$.* Let set Ω denote a set of all circuits existing in the digraph $(V, A, \varnothing)$.

We will treat two possible cases, I and II.

Case I. Let set Ω be empty; $\Omega = \varnothing$.

The unit-time scheduling problem $G_cMPT|[r_i], t_{ij} = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ may be constructed based on Algorithm 1, which follows.

Case II. Let set Ω be not empty; $\Omega \neq \varnothing$.

The mixed graph $G = (V, A, E)$ is colourable. So, due to Theorem 1, each circuit $(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}, v_{k_1})$ in set Ω has no vertices that are adjacent in the graph $(V, \varnothing, E)$. Therefore, all tasks of the set $\left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\} = V(k)$ must be processed simultaneously in the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$ with the circuit $(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}, v_{k_1})$ existing in the directed subgraph $(V, A, \varnothing)$. Assume that $\Omega = \bigcup\limits_{k=1}^{w} V(k) = \bigcup\limits_{k=1}^{w} \left\{(v_{k_1}, v_{k_2}, v_{k_3}, \ldots, v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}, v_{k_1})\right\}$. We delete arcs A_c determined in (4) from the mixed graph $G = (V, A, E)$ and apply Algorithm 1 to the constructed mixed graph $G^0 = (V, A \backslash A_c, E)$. Thus, we construct the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G^0 = (V, A \backslash A_c, E)$, which is equivalent to the problem of finding the optimal coloring $c(G^0)$. This unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$ is equivalent to the problem of finding the optimal colouring $c(G)$ of the mixed graph G. □

Algorithm 1: Determining the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$, which is equivalent to the problem of finding an optimal colouring $c(G)$.

```
Input:  A mixed graph $G = (V, A, E)$ with a circuit-free digraph $(V, A, \varnothing)$.
Output: The unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$ that is
        equivalent to the problem of finding the optimal colouring $c(G)$.
Step 1: Partition graph $(V, \varnothing, E)$ into maximal number of the connected components: $(V, \varnothing, E)$ =
        $(V_1, \varnothing, E_1) \cup \ldots \cup V_m, \varnothing, E_m) \cup (V_{m+1}, \varnothing, \varnothing) \ldots \cup (V_{m+r}, \varnothing, \varnothing)$, where subgraphs
        $(V_k, \varnothing, E_k)$ are connected components of the graph $(V, \varnothing, E)$ for $k \in \{1, 2, \ldots, m\}$, $|V_k| \geq 2$,
        and each subgraph $(V_j, \varnothing, \varnothing)$ determines an isolated vertex for index
        $j \in \{m+1, m+2, \ldots, m+r\}$. Denote isolated vertices as follows: $\left\{v_{j_1}\right\} = V_j$. Set $M = \varnothing$,
        $k = 1$, $i = 0$, and $l_0 = 0$.
Step 2: IF $k = m + 1$ GOTO step 5 ELSE find all maximum complete vertex-induced subgraphs
        $(V_k^1, \varnothing, E_k^1), \ldots, (V_k^{l_k}, \varnothing, E_k^{l_k})$ of the graph $(V_k, \varnothing, E_k)$. Set $r = 1$, $i := i + l_{k-1} + 1$.
Step 3: FOR every index $i$, add the new processor $M_i$ to the already constructed set;
        $M := M \cup \{M_i\}$. State that all tasks of the clique $V_k^r$ of the graph $(V_k, \varnothing, E_k)$ will be
        processed on the processor $M_i$; $V_k^r$ = $V_i$ = $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$, where all tasks
        $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ will be processed on the processor $M_i$. Set $i := i + 1$.
Step 4: IF $i = \sum_{h=0}^{k} l_h$ THEN $k := k + 1$ GOTO step 2 ELSE $r := r + 1$ GOTO step 3.
Step 5: FOR every index $j \in \{m+1, \ldots, m+r\}$, add the new processor $M_{i+j-m}$ to the already
        constructed set $M$; $M := M \cup \left\{M_{i+j-m}\right\}$. State that task $v_{j_1}$, $V_j = \left\{v_{j_1}\right\}$, which is isolated
        in the graph $(V, \varnothing, E)$, will be processed on processor $M_{i+j-m}$. Processor $M_{i+j}$ will
        process only task $v_{j_1}$. Set $M := M \cup \{M_{i+1}, \ldots, M_{i+r}\}$.
Step 6: FOR every arc $(v_p, v_q) \in A$, for which the implication (2) does not hold, determine the
        weak precedence constraint $v_p \to v_q$.
Step 7: FOR every arc $(v_p, v_q) \in A$, for which the implication (2) holds, determine the precedence
        constraint $v_p \Rightarrow v_q$.
Step 8: The unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$ is
        constructed, where the precedence constraints on the set $V$ are defined at steps 6 and 7.
        The set $M$ of the processors is defined at steps 3 and 5. STOP.
```

The above Algorithm 1 shows us how to construct the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$, which is equivalent to the problem of finding an optimal coloring $c(G)$ of the mixed graph $G = (V, A, E)$, all jobs in the set J being simple.

To illustrate Theorem 6, we solve Example 2 of the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G = (V, A, E)$ depicted in Figure 2.

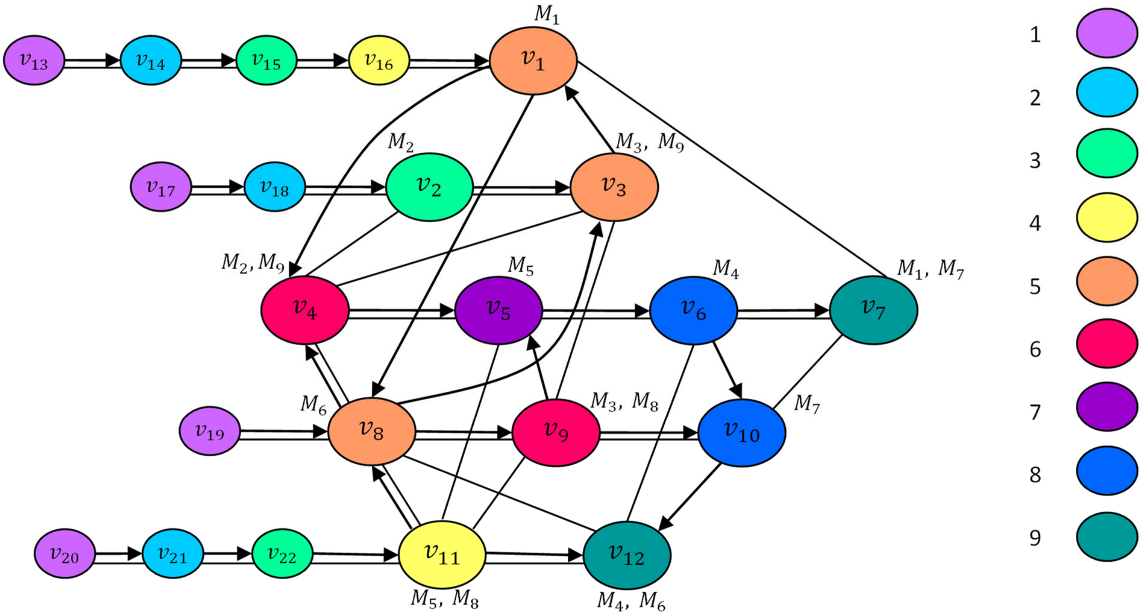

Figure 2. Mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$ determining input data for Example 2.

Example 2. *This example is similar to Example A2 with the input data shown in Figure A2; see Appendix B. The only difference is that the input data for Example 2 include the following release dates for the jobs* $\{J_1, J_2, \ldots, J_5\} = J$: $r_1 = 4$, $r_2 = 2$, $r_3 = 0$, $r_4 = 1$, $r_5 = 3$. *We determine the input data for Example 2 based on a mixed graph* $G^{**} = (V^{**}, A^{**}, E^{**})$ *that includes the subgraph* $G = (V, A, E)$ *presented in Figure 2. The release date* $r_1 = 4$ *of the job* $J_1 \in J$ *is represented by the union of path* $(v_{13}, v_{14}, v_{15}, v_{16}, v_1)$ *in the directed graph* $(V^{**}, A^{**}, \varnothing)$ *and chain* $(v_{13}, v_{14}, v_{15}, v_{16}, v_1)$ *in the graph* $(V^{**}, \varnothing, E^{**})$. *The release date* $r_2 = 2$ *of the job* $J_2 \in J$ *is represented by the union of path* (v_{17}, v_{18}, v_2) *in the directed graph* $(V^{**}, A^{**}, \varnothing)$ *and chain* (v_{17}, v_{18}, v_2) *in the graph* $(V^{**}, \varnothing, E^{**})$. *The release date* $r_4 = 1$ *of the job* $J_4 \in J$ *is represented by the union of arc* (v_{19}, v_8) *in the directed graph* $(V^{**}, A^{**}, \varnothing)$ *and edge* $[v_{19}, v_8]$ *in the graph* $(V^{**}, \varnothing, E^{**})$. *The release date* $r_5 = 3$ *of the job* $J_5 \in J$ *is represented by the union of path* $(v_{20}, v_{21}, v_{22}, v_{11})$ *in the directed graph* $(V^{**}, A^{**}, \varnothing)$ *and chain* $(v_{20}, v_{21}, v_{22}, v_{11})$ *in the graph* $(V^{**}, \varnothing, E^{**})$.

The directed subgraph $(V^{**}, A^{**}, \varnothing)$ of the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$ contains the single circuit (v_1, v_8, v_3, v_1). Since this circuit has no vertices that are adjacent in the subgraph $(V^{**}, \varnothing, E^{**})$ of the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$, there exists a semiactive schedule for Example 2 of the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$; see Theorem 4. One can construct the following optimal colouring of the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$ depicted in Figure 2: $c(v_1) = 5$ $c(v_2) = 3$, $c(v_3) = 5$, $c(v_4) = 6$, $c(v_5) = 7$, $c(v_6) = 8$, $c(v_7) = 9$, $c(v_8) = 5$, $c(v_9) = 6$, $c(v_{10}) = 8$, $c(v_{11}) = 4$, $c(v_{12}) = 9$, $c(v_{13}) = 1$, $c(v_{14}) = 2$, $c(v_{15}) = 3$, $c(v_{16}) = 4$, $c(v_{17}) = 1$, $c(v_{18}) = 2$, $c(v_{19}) = 1$, $c(v_{20}) = 1$, $c(v_{21}) = 2$, $c(v_{22}) = 3$.

The colouring $c(G)$ with $\chi(G) = 9$ is optimal for Example 2 since digraph $(V^{**}, A^{**}, \varnothing)$ contains a path $(v_{20}, v_{21}, v_{22}, v_{11}, v_8, v_4, v_5, v_6, v_7)$ and the graph $(V^{**}, \varnothing, E^{**})$ contains a chain $(v_{20}, v_{21}, v_{22}, v_{11}, v_8, v_4, v_5, v_6, v_7)$ whose lengths are equal to 9. Because of this path and chain, the inequality $\chi(G^{**}) \geq 9$ holds. Based on Theorem 6, the colouring $c(G^{**})$ defines the following optimal semiactive schedule for Example 2:

$$\begin{aligned} S = (&C(Q_{1,1}) = 5, C(Q_{2,1}) = 3, C(Q_{2,2}) = 5, C(Q_{3,1}) = 6, C(Q_{3,2}) = 7, C(Q_{3,3}) = 8, C(Q_{3,4}) = 9, \\ &C(Q_{4,1}) = 5, C(Q_{4,2}) = 6, C(Q_{4,3}) = 8, C(Q_{5,1}) = 4, C(Q_{5,2}) = 9) \end{aligned}$$

The correspondence between operations Q and vertices V is based on Remark 1. Due to Theorem 6, the following equalities hold: $\chi(G^{**}) = 9 = C_{\max}$.

4.2. Equivalent Scheduling Problems for Minimising either Makespan or Maximal Lateness

In [12], it was proven that the scheduling problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ reduces to the scheduling problem $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$ with the objective of minimising the maximal lateness $L_{\max} = \max\{C_i - D_i : J_i \in J\}$, where the given due dates $D_i \geq 0$ are integers.

Theorem 7 ([12]). *Every solvable unit-time problem* $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ *on the mixed graph G with integer release dates* $r_i \geq 0$ *reduces to unit-time problem* $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$ *with non-negative integer due dates, and vice versa.*

For illustrating Theorem 7, we solve Example 2 of the unit-time problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$ on the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$ shown in Figure 2 and the following Example 3 of the unit-time problem $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$.

Example 3. *In Example 3 of the problem* $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$ *on the mixed graph* $G = (V, A, E)$ *depicted in Figure 3, the set of jobs* J_1, J_2, J_3, J_4 *and* J_5 *are ready for processing on the processors* $M = \{M_1, M_2, \ldots, M_9\}$ *from a release date of zero. Each processor* $M_k \in M$ *must process the same subset of jobs as in Example 2. However, the orders for processing the same operations are opposite to the linear orders determined in Example 2. If the precedence constraints* $v_i \to v_j$ *and* $v_p \Rightarrow v_q$ *are given between operations belonging to different jobs in Example 2, the inverse precedence constraints* $v_j \to v_i$ *and* $v_q \Rightarrow v_p$ *must be given in Example 3 of the general shop scheduling problem* $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$. *For Example 3, the due dates of the given jobs* J_1, J_2, J_3, J_4, J_5 *are defined as follows:* $D_1 = \max\{r_j : J_j \in J\} - r_1 = 4 - 4 = 0$, $D_2 = 4 - r_2 = 4 - 2 = 2$, $D_3 = 4 - r_3 = 4 - 0 = 4$, $D_4 = 4 - r_4 = 4 - 1 = 3$, $D_5 = 4 - r_5 = 4 - 3 = 1$.

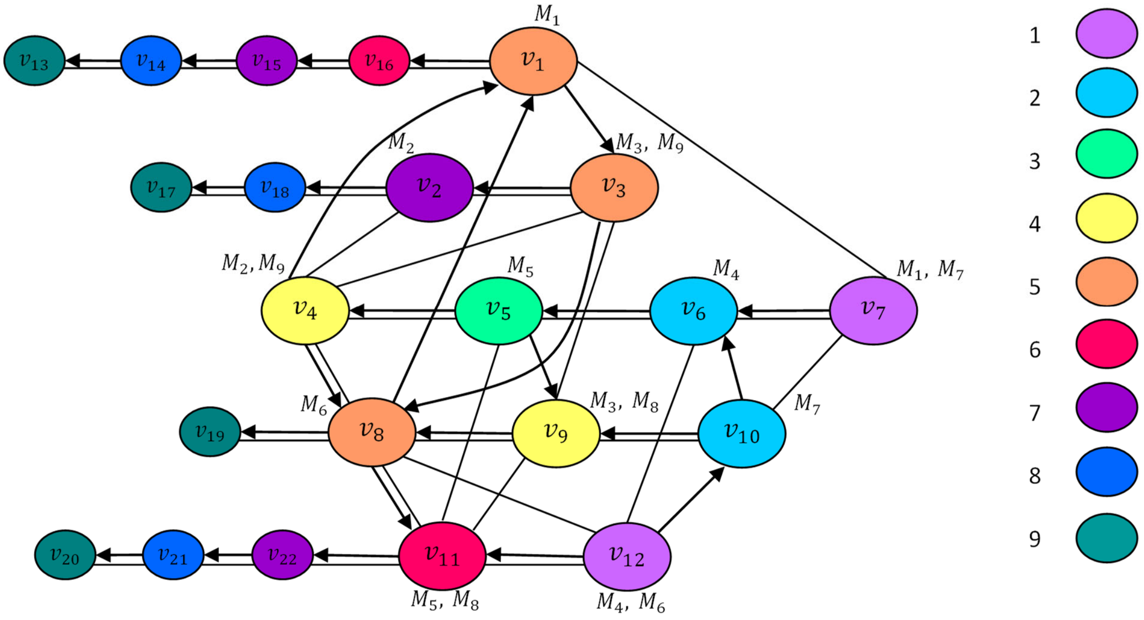

Figure 3. The colouring $c(G^{\leftarrow})$ determining the optimal semiactive schedule for the unit-time problem $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$.

The input data for Example 3 are given by the mixed graph $G^{\leftarrow} = (V^{**}, A^{\leftarrow}, E^{**})$ presented in Figure 3. The set V^{**} of vertices and the set E^{**} of edges in the mixed graph $G^{\leftarrow}$ are the same as those depicted in Figure 3 for the mixed graph G.

Compare the mixed graph depicted in Figure 3 for the problem $G_cMPT\big|t_{ij} = 1, [D_i]\big|L_{\max}$ and the mixed graph $G^{**} = (V^{**}, A^{**}, E^{**})$ depicted in Figure 2 for the problem $G_cMPT\big|[r_i], t_{ij} = 1\big|C_{\max}$. By Theorems 6 and 7, for Example 3 of the general shop unit-time scheduling problem $GcMPT|t_i = 1, [D_i]|L_{\max}$ on the mixed graph $G^{\leftarrow} = (V^{**}, A^{\leftarrow}, E^{**})$, the optimal semiactive schedule S is determined by the optimal colouring of mixed graph $G^{\leftarrow} = (V^{**}, A^{\leftarrow}, E^{**})$ shown in Figure 3: $c(v_1) = 5$ $c(v_2) = 7$, $c(v_3) = 5$, $c(v_4) = 4$ $c(v_5) = 3$, $c(v_6) = 2$, $c(v_7) = 1$, $c(v_8) = 5$, $c(v_9) = 4$, $c(v_{10}) = 2$, $c(v_{11}) = 6$, $c(v_{12}) = 1$, $c(v_{13}) = 9$, $c(v_{14}) = 8$, $c(v_{15}) = 7$, $c(v_{16}) = 6$, $c(v_{17}) = 9$, $c(v_{18}) = 8$, $c(v_{19}) = 6$, $c(v_{20}) = 9$, $c(v_{21}) = 8$, $c(v_{22}) = 7$.

The colouring $c(G^{\leftarrow})$ defines the following optimal schedule for Example 3:

$$S = (C(Q_{1,1}) = 5, C(Q_{2,1}) = 7, C(Q_{2,2}) = 5, C(Q_{3,1}) = 4, C(Q_{3,2}) = 3, C(Q_{3,3}) = 2, C(Q_{3,4}) = 1,$$
$$C(Q_{4,1}) = 5, C(Q_{4,2}) = 4, C(Q_{4,3}) = 2, C(Q_{5,1}) = 6, C(Q_{5,2}) = 1)$$

The optimal maximum lateness is equal to $\chi(G^{\leftarrow}) - \max\{r_i : J_i \in J\} = 9 - 4 = 5$.

5. An Optimal Schedule of Interruptible Operations with Integer Processing Times

We investigate the scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with integer release dates, positive integer operation durations, and allowed operation interruptions. In the three-field notation, *pmtn* indicates allowed preemptions of the operations and $[t_{ij}]$ ($[r_i]$, respectively) means that the operation durations (release dates of the jobs) are positive (non-negative) integers.

5.1. Partitions of the Interruptible Operations with Integer Duration into Unit-Time Operations

If interruptions of the operation are allowed, then a set of schedules is usually larger than a set of schedules of non-interruptible operations of the set Q. The larger set of schedules may complicate optimal scheduling [11]. Obviously, allowing interruptions of the operations from the set Q may decrease an optimal value of the objective function. Thus, for most scheduling problems, it is desirable to reduce the number of allowed interruptions of the operations Q without losing an optimal semiactive schedule.

Lemma 2. *If the release dates r_i of the jobs $J_i \in J$ are non-negative integers and the operations $Q_{ij} \in Q$ have positive integer durations $t_{ij} > 0$, then there exists an optimal semiactive schedule such that interruptions in processing operations occur only at the integer times.*

Proof. The interruption of the operation $Q_{ij} \in Q$ can reduce the value of a regular objective function in the semiactive schedule without interruptions only if the interruption is at a completion time of at least one operation $Q_{ij} \in Q$. In fact, due to such interruption of the operation $Q_{ij} \in Q$ processed on the processor $M_{\mu(i.j)} \in M$, another operation $Q_{uv} \in Q$ with $u \neq i$ can start its processing on the same processor $M_{\mu(i.j)} = M_{\mu(u,v)}$, since this processor becomes free.

Since the release dates r_i of the given jobs $J_i \in \left\{J_1, J_2, \ldots, J_{|J|}\right\}$ are non-negative integers and operations have positive integer durations, in any semiactive schedule without operation interruptions, the realisation of each operation can be completed only at integer times. Lemma is proven. □

Due to Lemma 2, operations $Q_{1,1} \in Q^{(1)}$ with positive integer durations can be partitioned into $t_{1,1}$ unit-time operations. We denote these unit-time operations as follows: $\{v_1, v_2, \ldots, v_{t_{1,1}}\}$. Let $\left\{v_{t_{1,1}+1}, v_{t_{1,1}+2}, \ldots, v_{t_{1,1}+t_{1,2}}\right\}$ be a set of all unit-time operations generated by similar partitioning the operation $Q_{1,2} \in Q^{(1)}$. Similarly, we partition operations $Q^{(1)} \backslash \{Q_{1,1}, Q_{1,2}\}$ into unit-time operations and assign sequential indexes to the generated unit-time operations.

Continuing the above partitioning procedure of operations from set $Q^{(1)}$ with integer durations, we generate the following set of unit-time operations:

$$W_1 = \left\{v_1, v_2, \ldots, v_{|W_1|}\right\}, \text{ where } |W_1| = \sum_{j=1}^{|Q^{(1)}|} t_{1,j}. \tag{5}$$

The generated unit-time operations will be ordered as follows: $(v_1, v_2, \ldots, v_{|W_1|})$ within any feasible schedule; see Lemma 2.

Remark 2. *We represent all unit-time operations generated by the above partition of operations Q as vertices in the mixed graph $G = (V, A, E)$. Here and elsewhere, the generated unit-time operations are linearly ordered in the ascending order of their indexes. For each job $J_i \in J$, the unit-time operations generated by the above partitioning procedure of the integer-time operations $Q^{(i)}$ will be fulfilled in the ascending order of their indexes in any semiactive schedule.*

We continue with the generated unit-time operation $v_{|W_1|+1}$ by sequentially indexing the generated unit-time operations into which integer-time operations from set $Q^{(2)} = \left\{Q_{2,1}, Q_{2,2}, \ldots, Q_{2,|Q^{(2)}|}\right\}$ contained in the job $J_2 \in J$ are partitioned. Thus, we obtain the following set of the unit-time operations:

$$W_2 = \left\{v_{|W_1|+1}, v_{|W_1|+2}, \ldots, v_{|W_2|}\right\}, \text{where } |W_2| = |W_1| + \sum_{j=1}^{|Q^{(2)}|} t_{2,j}, \tag{6}$$

of the linearly ordered unit-time operations generated from integer-time operations of job J_2; see Remark 2. Using the above partitioning technique, we index the unit-time operations of the jobs in set $J \setminus \{J_1, J_2\}$ in the increasing order of job indexes and indexes of all generated unit-time operations. At the end of this partitioning process, for operations with integer job durations $J_{|J|} \in J$, the following set is obtained:

$$W_{|J|} = \left\{v_{|W_{|J|-1}+1}, v_{|W_{|J|-1}+2}, , \ldots, v_{|W_{|J|}|}\right\}, \text{ where } |W_{|J|}| = |W_{|J|-1}| + \sum_{j=1}^{|Q^{(|J|)}|} t_{|J|,j}, \tag{7}$$

of the ordered unit-time operations generated by the integer-time operations belonging to the job $J_{|J|}$. All jobs from the set J will be processed based on Remark 2 applied to the set $W := \cup_{i=1}^{|J|} W_i$ of all generated unit-time operations. Furthermore, for each job $J_i \in J$, the unit-time operations of the set W_i generated by the above partitioning procedure applied to the integer-time operations of the set $Q^{(i)}$ are fulfilled in increasing order of their indexes during the realisation of each semiactive schedule.

5.2. Relationships between Scheduling Problems and Optimal Mixed Graph Colourings

We next prove Theorem 8, which generalizes Theorem 5 to the problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ of constructing an optimal schedule for fulfilling jobs $J_i \in J$ with integer release dates $r_i \geq 0$ and positive durations $t_{ij} > 0$ of the operations $Q_{ij} \in Q^{(i)} \subseteq Q$ with allowed interruptions.

Theorem 8. *Every solvable scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ pseudo-polynomially reduces to a problem of constructing an optimal colouring of vertices of the mixed graph $G = (V, A, E)$. For every colourable mixed graph G, there exists a scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ on the mixed graph G, which is equivalent to the problem of finding the optimal colouring $c(G)$ of vertices of the same mixed graph G.*

Proof. We consider arbitrary solvable scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ of constructing a makespan-optimal schedule for fulfilling jobs $J = \left\{J_1, J_2, \ldots, J_{|J|}\right\}$ by the available dedicated processors $M = \left\{J_1, J_2, \ldots, J_{|M}\right\}$. We show how to construct the mixed graph $G = (V, A, E)$ such that optimal colouring $c(G)$ defines a solution to the problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$.

Due to Lemma 2, there exists an optimal semiactive schedule such that interruptions in the operations (if any) occur only at the integer time moments. Thus, to construct the mixed graph $G = (V, A, E)$, we will determine a set of unit-time operations

$$W = \left\{v_1, v_2, \ldots, v_{|W|}\right\} = \cup_{i=1}^{|J|} W_i$$

generated by the above partitioning of the integer-time operations belonging to jobs $J_i \in J = \left\{J_1, J_2, \ldots, J_{|J|}\right\}$ into the unit-time tasks sequentially, as shown in Section 5.1. The Formula (5) determines the set W_1 of the unit-time tasks generated by integer-time tasks $Q^{(1)}$ belonging to a job J_1.

The Formula (6) determines the set W_2 of the unit-time tasks generated by the integer-time tasks $Q^{(2)}$ belonging to the job J_2.

Similar formulas for sets $Q^{(3)}, Q^{(4)}, \ldots, Q^{(|J|-1)}$ of the integer-time tasks sequentially determine sets $W_3, W_4, \ldots, W_{|J|-1}$, respectively, of the generated unit-time tasks.

Finally, the Formula (7) determines the set $W_{|J|}$ of the unit-time tasks generated by the integer-time tasks $Q^{(|J|)}$ belonging to the last job $J_{|J|} \in J$.

The above partitioning(see Section 5.1) of the integer-time tasks $Q = \underset{i=1}{\overset{|J|}{\cup}} Q^{(i)}$ into the unit-time tasks determines a subgraph $(W, \varnothing, \varnothing)$ with an empty set of arcs and an empty set of edges in the indicated mixed graph $G = (V, A, E)$, where $W \subset V$. Based on Remark 2, every job $J_i \in J$ consisting of the ordered set of the integer-time tasks $Q^{(i)}$ is partitioned into the linearly ordered set of the unit-time tasks of the following form: $W_i = \left\{v_{|W_{i-1}|+1}, v_{|W_{i-1}|+2}, \ldots, v_{|W_i|}\right\}$.

In any semiactive schedule, the unit-time tasks W_i have to be executed in ascending order of indexes of the unit-time tasks $v_k \in W_i$; see Remark 2. By Definition 1, the desired mixed graph $G = (V, A, E)$ must contain the set of arcs $A_i = \{(v_{|W_{i-1}|+1}, v_{|W_{i-1}|+2}),$ $(v_{|W_{i-1}|+2}, v_{|W_{i-1}|+3}), \ldots, (v_{|W_i|-1}, v_{|W_i|})\}$ and the following set of edges: $E_i = \left\{[v_{|W_{i-1}|+1}, v_{|W_{i-1}|+2}], [v_{|W_{i-1}|+2}, v_{|W_{i-1}|+3}], \ldots, [v_{|W_i|-1}, v_{|W_i|}]\right\}$.

The sets of arcs A_i and edges E_i will determine the linear order for the execution of the unit-time tasks from set W_i when implementing any semiactive schedule for the problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$; see Remark 2.

Thus, we constructed a subgraph $(W, \cup_{i=1}^{|J|} A_i, \cup_{i=1}^{|J|} E_i)$ of the desired mixed graph $G = (V, A, E)$ where inclusion $W \subset V$ holds. For every processor $M_k \in M$, set Q_k of all integer-time tasks fulfilled by the processor M_k defines set V_k of unit-time tasks fulfilled by the processor M_k. The cardinality of set V_k is equal to the total processing time of the integer-time tasks for fulfilling all jobs in the set $J^{(k)} =: \left\{J_{k_1}, J_{k_2}, \ldots, J_{|J^{(k)}|}\right\}$. So, the following equality holds:

$$\left|V_k\right| = \sum\nolimits_{J_i \in J^{(k)}} t_{ik} \tag{8}$$

Set V_k is partitioned into $\left|J^{(k)}\right|$ subsets V_k^j of the tasks with unit durations for fulfilling jobs $J_{k_j} \in J^{(k)}$, i.e., the equality $V_k = V_k^1 \cup V_k^2 \cup \ldots \cup V_k^{|J^{(k)}|}$ holds, where $V_k^j \neq \varnothing$ and $V_k^j \cap V_k^h = \varnothing$ for each indices $j \neq h$. As can be easily verified, the forbiddance to simultaneously execute every pair of the unit-time tasks of the set V_k in a semiactive schedule is determined by the complete $\left|J^{(k)}\right|$-partite subgraph $(V_k, \varnothing, E_k)$ of the mixed graph $G = (V, A, E)$ with the following set $\left\{V_k^1, V_k^2, \ldots, V_k^{|J^{(k)}|}\right\}$ of vertices and the following edge set E_k with the cardinality determined as follows:

$$|E_k| = \sum_{i=1}^{|J^{(k)}|-1} \sum\nolimits_{j=i+1}^{|J^{(k)}|} \left\{\left|V_k^i\right| \times \left|V_k^j\right|\right\} \tag{9}$$

Together with the precedence constraints determined above between tasks belonging to the same job $J_i \in J$, the input data for the problem $G_cMPT\big|[r_i],[t_{ij}],pmtn\big|C_{\max}$ may include the precedence constraints defined on the set of tasks, which belong to different jobs of the set J. Let these precedence constraints be determined as follows:

$$R_{\Rightarrow} = \left\{v_{k_1} \Rightarrow v_{k_2},\ v_{k_3} \Rightarrow v_{k_4}, \dots, v_{k_{|k|}-1} \Rightarrow v_{k_{|k|}}\right\} \tag{10}$$

Moreover, the following weak precedence constraints on the set of tasks, which belong to different jobs of the set J, may also be determined:

$$R_{\rightarrow} = \left\{v_{h_1} \rightarrow v_{h_2},\ v_{h_3} \rightarrow v_{h_4}, \dots, v_{h_{|h|}-1} \rightarrow v_{h_{|h|}}\right\} \tag{11}$$

According to Definition 1, we introduce the precedence constraints (10) by adding the set $A_{|J|+1} = \left\{(v_{k_1}, v_{k_2}), (v_{k_2}, v_{k_3}), \dots, (v_{k_{|k|}-1}, v_{k_{|k|}})\right\}$ of the arcs and the set $E_{|J|+1} = \left\{[v_{k_1}, v_{k_2}], [v_{k_2}, v_{k_3}], \dots, [v_{k_{|k|}-1}, v_{k_{|k|}}]\right\}$ of the edges in the mixed graph $G = (V, A, E)$.

To determine the weak precedence constraints (11), it is sufficient to add the following set $A_{|J|+2} = \left\{(v_{h_1}, v_{h_2}), (v_{h_2}, v_{h_3}), \dots, (v_{h_{|h|}-1}, v_{h_{|h|}})\right\}$ of the arcs in the desired mixed graph $G = (V, A, E)$. Let the input data for the scheduling problem $G_cMPT\big|[r_i],[t_{ij}],pmtn\big|C_{\max}$ include the subsets $V(1), V(2), \dots, V(l)$ of the unit-time operations from the set Q such that operations of the subset $V(g) = \left\{v_{g_1}, v_{g_2}, \dots, v_{g_{|V(g)|}}\right\} \subset Q$ must be executed simultaneously in any schedule, where $g \in \{1, 2, \dots, l\}$. Therefore, the set of weak precedence constraints $R_{\rightarrow}$ determined in (11) must contain the following subset:

$$\cup_{g=1}^{l}\left\{v_{g_1} \rightarrow v_{g_2}, v_{g_2} \rightarrow v_{g_3}, \dots, v_{g_{|g|}-1} \rightarrow v_{g_{|g|}}, v_{g_{|g|}} \rightarrow v_{g_1}\right\}$$

It is clear that the above constructed set $A_{|J|+2}$ of the arcs must contain the following set of the arcs: $A_{|J|+3} = \cup_{g=1}^{l}\left\{(v_{g_1}, v_{g_2}), (v_{g_2}, v_{g_3}), \dots, (v_{g_{|g|}-1}, v_{g_{|g|}}), (v_{g_{|g|}}, v_{g_1})\right\}$; see Definition 1. We have constructed the arc set $A^o = \left\{\cup_{i=1}^{|J|} A_i\right\} \cup A_{|J|+1} \cup A_{|J|+2} \cup A_{|J|+3}\Big\}$, the edge set $E^o = \left\{\cup_{i=1}^{|J|} E_i\right\} \cup E_{|J|+1}$ and the vertex set $W = \left\{v_1, v_2, \dots, v_{|W|}\right\} = \cup_{i=1}^{|J|} W_i$, where each vertex determines the unit-time operation (task) generated by the operation from the set Q with integer durations. Thus, we have constructed the mixed graph (W, A^o, E^o), which is a subgraph of the desired mixed graph G.

The input data for the considered solvable problem $G_cMPT\big|[r_i],[t_{ij}],pmtn\big|C_{\max}$ include also the non-negative integer release dates $r_1, r_2, \dots, r_{|J|}$ of the jobs $J_1, J_2, \dots, J_{|J|}$, respectively. Therefore, for every job $J_i \in J$, the set $W(r_i) = \left\{v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}}\right\}$ of the vertices and the path $(v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}})$, whose length is equal to $r_i \geq 0$, are determined. We add all these vertices and arcs of the following paths: $(v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}})$ to subgraph $(W, A^o, \varnothing)$ of mixed graph (W, A^o, E^o).

We also add the edges of the following chains $(v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}})$ to subgraph $(W, \varnothing, E^o)$ of mixed graph (W, A^o, E^o). Let $G = \{V, A, E)$ be a mixed graph generated by the mixed graph (W, A^o, E^o) after adding all the vertices $\overset{|J|}{\underset{i=1}{\cup}} W(r_i)$, all the arcs of paths $(v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}})$ and all the edges of chains $(v_{i^0_1}, v_{i^0_2}, \dots, v_{i^0_{r_{i-1}}})$.

Thus, we have constructed the desired mixed graph $G = (V, A, E)$. For the constructed mixed graph, the following equality $V = W + \overset{|J|}{\underset{i=1}{\cup}} W(r_i)$ holds. Due to the above construction of the mixed graph, the precedence constraints defined on the set of tasks for the problem $G_cMPT\big|[r_i],[t_{ij}],pmtn\big|C_{\max}$ are determined by graph (W, A^o, E^o). The prohibition of simultaneously fulfilling every pair of unit-time tasks of the set $V^{(k)}$ in any semiactive schedule is determined by the complete $\left|J^{(k)}\right|$-partite graph $(V^{(k)}, \varnothing, E^{(k)})$ with the vertex set $\left\{V_k^1, V_k^2, \ldots, V_k^{|J^{(k)}|}\right\}$. The given release dates $r_1, r_2, \ldots, r_{|J|}$ of the given jobs $J_1, J_2, \ldots, J_{|J|}$ are defined by a subgraph $(\bigcup_{i=1}^{|J|} W(r_i), A\backslash A^o, E\backslash E^o)$ of the constructed mixed graph $G = (V, A, E)$.

As can be easily verified, the constructed mixed graph $G = (V, A, E)$ completely defines the input data of the considered scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$. Hence, for the solvable general scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ determined on mixed graph $G = (V, A, E)$, there exists a semiactive schedule

$$S = (C(v_1), C(v_2), \ldots, C(Q_{|W|}) \tag{12}$$

which defines the colouring $c(G)$ of the constructed mixed graph $G = (V, A, E)$ as follows: the equality $c(v_i) = C(v_i)$ holds for every vertex $v_i \in V$. It is clear that the optimal semiactive schedule S that exists for the solvable scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ on the mixed graph $G = (V, A, E)$ defines the optimal colouring $c(G)$ of the constructed mixed graph $G = (V, A, E)$.

Compare the form (1) of semiactive schedules existing for unit-time problems and the form (12) of semiactive schedules existing for problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ with the aim to construct a makespan-optimal schedule for fulfilling jobs $J_i \in J$ with the positive integer processing times $t_{ij} > 0$ of the operations $Q_{ij} \in Q$, provided that operation interruptions are allowed. Along with the completion times of the given operations $Q_{ij} \in Q^{(i)} \subseteq Q$, with integer processing times, the latter semiactive schedule must define the completion times of the unit-time operations generated by the above partitioning procedure of the operations $Q_{ij} \in Q^{(i)}$ with integer processing times. Due to the equalities (8) and (9), the described reduction of the general scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ to the problem of finding the optimal colouring $c(G)$ of the constructed mixed graph $G = (V, A, E)$ is pseudo-polynomial.

The proof of the first part of Theorem 8 is completed.

It should be noted that an optimal semiactive schedule existing for the problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ with positive integer durations of all given operations, integer non-negative release dates of the given jobs, and allowed operation interruptions can be obtained from the optimal semiactive schedule existing for the unit-time scheduling problem $G_cMPT\left|[r_i], t_{ij} = 1\right|C_{\max}$, and vice versa. Indeed, if the integer-time operations $Q_{ij} \in Q$ are partitioned into the unit-time operations W based on the above partitioning procedure described in Section 5.1 (see Lemma 2 and Remark 2), the operation interruptions of the set W cannot reduce the value $C_{\max}$ for the optimal semiactive schedule. Thus, the possibilities of interrupting the operations in the scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ can be ignored when constructing a makespan-optimal semiactive schedule that exists for the considered scheduling problem.

Hence, the problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ can be solved as a special case of the unit-time scheduling problem $G_cMPT\left|[r_i], t_{ij} = 1\right|C_{\max}$; see Section 4.1. Therefore, the proof of the second part of Theorem 8 follows directly from the second part of Theorem 6. Theorem 8 is proved. □

6. Example 4 of General Scheduling Problem with Interruptible Operations

To illustrate Theorem 8, we describe Example 4 of solvable scheduling problem $G_cMPT\left|[r_i], [t_{ij}], pmtn\right|C_{\max}$ with a set of jobs $J = \{J_1, J_2, J_3, J_4\}$, seven dedicated processors $M = \{M_1, M_2, \ldots, M_7\}$, integer release dates $r_i \geq 0$, integer processing times $t_{ij} > 0$ of the multi-processor tasks, two types of the precedence constraints between operations, and allowed interruptions of the integer-time operations.

Example 4. *The main part of the input data for Example 4 is presented in* Table 1. *Other input data for Example 4 will be determined after constructing a mixed graph with a set W of the vertices corresponding to unit-time operations generated by partitioning the integer-time operations $Q_{ij} \in Q$ based on Remark 2.*

Table 1. A main part of the input data for Example 4.

Operations $Q_{1,j}$ of the job J_1	$Q_{1,1}$	$Q_{1,2}$	-	-
Processors $M_{\mu(1,j)}$	M_1, M_2	M_7	-	-
Duration $t_{1,j}$ of operation $Q_{1,j}$	2	1	-	-
Operations $Q_{2,j}$ of the job J_2	$Q_{2,1}$	$Q_{2,2}$	$Q_{2,3}$	-
Processors $M_{\mu(2,j)}$	M_3, M_7	M_1	M_2,M_6	-
Duration $t_{2,j}$ of operation $Q_{2,j}$	2	1	3	-
Operations $Q_{3,j}$ of the job J_3	$Q_{3,1}$	$Q_{3,2}$	$Q_{3,3}$	$Q_{3,4}$
Processors $M_{\mu(2,j)}$	M_3	M_1	M_4	M_5,M_6
Duration $t_{3,j}$ of operation $Q_{3,j}$	1	1	2	2
Operations $Q_{4,j}$ of the job J_4	$Q_{4,1}$	$Q_{4,2}$	-	-
Processors $M_{\mu(4,j)}$	M_1, M_3	M_4,M_5	-	-
Duration $t_{4,j}$ of operation $Q_{4,j}$	1	3	-	-

Using the constructive proof of the first part of Theorem 8, we next show how to construct the mixed graph determining the input data of Example 4. Based on Lemma 2, we first partition the integer-time operations $Q_{1,j} \in Q^{(1)}$ into operations $W_1 = \{v_1, v_2, v_3\}$ with unit durations. Similarly, we partition the integer-time operations $Q_{2,j} \in Q^{(2)}$ into operations $W_2 = \{v_4, v_5, v_6, v_7, v_8, v_9\}$ with unit durations. Then, we partition the integer-time operations $Q_{3,j} \in Q^{(3)}$ into operations $W_3 = \{v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$ with unit durations. Finally, we partition the operations $Q_{4,j} \in Q^{(4)}$ of the last job into the operations $W_4 = \{v_{16}, v_{17}, v_{18}, v_{19}\}$ with unit durations. As a result, we construct the following set of operations: $W = \{v_1, v_2, \ldots, v_{19}\} = \cup_{i=1}^{4} W_i$ with unit durations generated from the integer-time operations of four given jobs $J = \{J_1, J_2, J_3, J_4\}$.

Due to Remark 2 and Definition 1, the mixed graph $G = (V, A, E)$ contains a set of arcs A_i and a set of edges E_i that determine the linear order of the operations of the set W_i in the semiactive schedule that exists for Example 4. The set of arcs A_i and the set of edges E_i must be constructed for each job $J_i \in \{J_1, J_2, J_3, J_4\}$. Thus, we have constructed a subgraph $(W, \cup_{i=1}^{|J|} A_i, \cup_{i=1}^{|J|} E_i)$ of mixed graph $G = (V, A, E)$ with inclusion $W \subset V$; see Figure 4, where the desired mixed graph G is depicted.

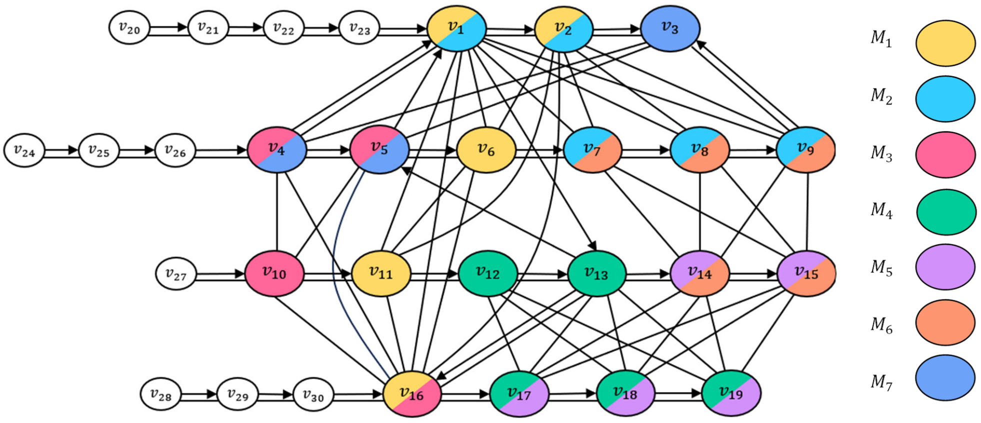

Figure 4. The mixed graph defining input data for Example 4 of the problem $G_c MPT \big| [r_i], [t_{ij}], pmtn \big| C_{\max}$.

For each processor $M_k \in M$, the operation set Q_k defines the set V_k of operations with unit durations to be executed on that processor M_k. The cardinality $|V_k|$ of set V_k is equal to the total processing time of the integer-time operations for fulfilling a set of the jobs $J^{(k)} = \left\{J_{k_1}, J_{k_2}, \ldots, J_{|J^{(k)}|}\right\}$, $k \in \{1, 2, \ldots, 7\}$. Based on the Formula (8), we calculate these cardinalities as follows:

$$|V_1| = \textstyle\sum_{J_i \in J^{(1)}} t_{i,1} = 2+1+1+1 = 5, |V_2| = \sum_{J_i \in J^{(2)}} t_{i,2} = 2+3 = 5,$$
$$|V_3| = \textstyle\sum_{J_i \in J^{(3)}} t_{i,3} = 2+1+1 = 4, |V_4| = \sum_{J_i \in J^{(4)}} t_{i,4} = 2+3 = 5,$$
$$|V_5| = \textstyle\sum_{J_i \in J^{(5)}} t_{i,5} = 2+3 = 5, |V_6| = \sum_{J_i \in J^{(6)}} t_{i,6} = 3+2 = 5,$$
$$|V_7| = \textstyle\sum_{J_i \in J^{(7)}} t_{i,7} = 1+2 = 3.$$

The prohibition of simultaneously executing each pair of unit-time operations of set V_k on the processor M_k in any semiactive schedule is determined by the complete $\left|J^{(k)}\right|$-partite subgraph $(V_k, \varnothing, E_k)$ with vertex set $\left\{V_k^1, V_k^2, \ldots, V_k^{|J^{(k)}|}\right\}$ as $\left|J^{(k)}\right|$ parts in the $\left|J^{(k)}\right|$-partite subgraph $(V_k, \varnothing, E_k)$ and edge set E_k, $k \in \{1, 2, \ldots, 7\}$.

Based on the Formula (9), we calculate the cardinalities $|E_k|, k \in \{1, 2, 3, 4\}$, as follows:

$$|E_1| = \sum_{i=1}^{|J^{(1)}|-1} \{\textstyle\sum_{j=i+1}^{|J^{(1)}|}\left\{\left|V_1^i\right| \times \left|V_1^j\right|\right\}\}= \{\{2 \times 1\} + \{2 \times 1\} + \{2 \times 1\}\} + \{\{1 \times 1\} + \{1 \times 1\}\} + \{\{1 \times 1\}\} = 9,$$
$$|E_2| = \sum_{i=1}^{|J^{(2)}|-1} \{\textstyle\sum_{j=i+1}^{|J^{(2)}|}\left\{\left|V_1^i\right| \times \left|V_1^j\right|\right\}\} = 2 \times 3 = 6, |E_3| = \sum_{i=1}^{|J^{(3)}|-1} \{\textstyle\sum_{j=i+1}^{|J^{(3)}|}\left\{\left|V_1^i\right| \times \left|V_1^j\right|\right\}\} = \{\{2 \times 1\} + \{2 \times 1\}\} + \{1 \times 1\} = 5,$$
$$|E_4| = |E_5| = |E_6| = |E_2| = 2 \times 3 = 6, |E_7| = 1 \times 2 = 2.$$

It is clear that, due to the 4-partite graph $(V_1, \varnothing, E_1)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_1 on the processor M_1. Due to the tripartite graph $(V_3, \varnothing, E_3)$, it is forbidden to simultaneously fulfill a pair of operations from the V_3 on the processor M_3. Due to the bipartite graph $(V_2, \varnothing, E_2)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_2 on the processor M_2. Due to the bipartite graph $(V_4, \varnothing, E_4)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_4 on the processor M_4. Due to the bipartite graph $(V_5, \varnothing, E_5)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_5 on the processor M_5. Due to the bipartite graph $(V_6, \varnothing, E_6)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_6 on the processor M_6. Due to the bipartite graph $(V_7, \varnothing, E_7)$, it is forbidden to simultaneously fulfill a pair of the operations of set V_7 on the processor M_7.

Let precedence constraints determined on a set of operations belonging to different jobs in Example 4 be given as follows: $R_{\Rightarrow} = \{v_4 \Rightarrow v_1,\ v_9 \Rightarrow v_3, v_{13} \Rightarrow v_{16},\}$. *Let the weak precedence constraints in Example 4 be given as follows:* $R_{\rightarrow} = \{v_1 \rightarrow v_{13},\ v_{13} \rightarrow v_5, v_5 \rightarrow v_1\}$.

To include the precedence constraints $R_{\Rightarrow}$ in the input data of Example 4, we add the set of arcs $A_{|J|+1} = \{(v_4, v_1), (v_9, v_3), (v_{13}, v_{16})\}$ and the edge set $E_{|J|+1} = \{[v_1, v_4), \{v_3, v_9], [v_{13}, v_{16}]\}$ to the mixed graph $G = (V, A, E)$. To include weak precedence constraints $R_{\rightarrow}$ in the input data of Example 4, we add the arc set $A_{|J|+2} = \{(v_1, v_{13}), (v_{13}, v_5), (v_5, v_1)\}$ to the mixed graph $G = (V, A, E)$. We see that set of arcs $A_{|J|+2}$ determines circuit (v_1, v_{13}, v_5, v_1) in the directed graph $(V, A, \varnothing)$. Therefore, unit-time operations of set $\{v_1, v_{13}, v_5\}$ have to be simultaneously fulfilled.

Let the following release dates of the jobs $J = \{J_1, J_2, J_3, J_4\}$ *be given as a part of the input data of Example 4:* $r_1 = 4$,$r_2 = 3$, $r_3 = 1$, $r_4 = 3$.

Release date $r_1 = 4$ of the job $J_1 \in J$ is determined by the union of path $(v_{20}, v_{21}, v_{22}, v_{23}, v_1)$ in the directed graph $(V, A, \varnothing)$ and chain $(v_{20}, v_{21}, v_{22}, v_{23}, v_1)$ in the subgraph $(V, \varnothing, E)$ of the mixed graph $G = (V, A, E)$. Release date $r_2 = 3$ is presented by the union of path $(v_{24}, v_{25}, v_{26}, v_4)$ in the directed graph $(V, A, \varnothing)$ and chain $(v_{24}, v_{25}, v_{26}, v_4)$ in the subgraph $(V, \varnothing, E)$ of the desired mixed graph $G = (V, A, E)$. Release date $r_3 = 1$ of the job $J_3 \in J$ is presented by the union of arc (v_{27}, v_{10}) in the directed graph $(V, A, \varnothing)$ and edge $[v_{27}, v_{10}]$ in the subgraph $(V, \varnothing, E)$ of the desired mixed graph $G = (V, A, E)$. Release date $r_4 = 3$ of the job $J_4 \in J$ is presented by the union of path $(v_{28}, v_{29}, v_{30}, v_{16})$ in the directed graph $(V, A, \varnothing)$ and chain $(v_{28}, v_{29}, v_{30}, v_{16})$ in the subgraph $(V, \varnothing, E)$ of the desired mixed graph $G = (V, A, E)$. Now, the desired mixed graph G is constructed. Figure 4 presents the constructed mixed graph G determining the whole input data for Example 4 of the problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with four jobs $J = \{J_1, J_2, J_3, J_4\}$ and seven dedicated processors $M = \{M_1, M_2, \dots, M_7\}$.

Due to Theorem 8, a solvable problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ (like Example 4) reduces to determining optimal colouring $c(G)$. An existing semiactive schedule for the scheduling problem with possible interruptions of the operations is determined by set (12) of completion times of the unit-time operations of set W. The semiactive schedule is determined by the colouring $c(G)$ of the mixed graph G, in which the equality $c(v_i) = C(v_i)$ holds for the vertices $v_i \in W$. The makespan-optimal semiactive schedule that exists for Example 4 is determined by the optimal colouring $c(G)$, where $c(v_1) = 5$, $c(v_2) = 10$, $c(v_3) = 11$, $c(v_4) = 4$, $c(v_5) = 5$, $c(v_6) = 6$, $c(v_7) = 7$, $c(v_8) = 8$, $c(v_9) = 9$, $c(v_{10}) = 2$, $c(v_{11}) = 3$, $c(v_{12}) = 4$, $c(v_{13}) = 5$, $c(v_{14}) = 6$, $c(v_{15}) = 7$, $c(v_{16}) = 7$, $c(v_{17}) = 8$, $c(v_{18}) = 9$, $c(v_{19}) = 10$. To show that the colouring $c(G)$ is optimal, we next prove the inequality $\chi(G) \geq 11$.

In the mixed graph G depicted in Figure 4, there is a path $(v_{24}, v_{25}, v_{26}, v_4, v_5, v_6, v_7, v_8, v_9, v_3)$ in the directed subgraph $(V, A, \varnothing)$ and a chain $(v_{24}, v_{25}, v_{26}, v_4, v_5, v_6, v_7, v_8, v_9, v_3)$ in the subgraph $(V, \varnothing, E)$. Therefore, a minimal colour of the vertex v_9 is not less than 9, $c(v_9) \geq 9$ and the minimal colour of the vertex v_3 is not less than 10; $c(v_3) \geq 10$. In mixed graph $G = (V, A, E)$, there is a path $(v_{20}, v_{21}, v_{22}, v_{23}, v_1, v_2, v_3)$ in the directed subgraph $(V, A, \varnothing)$ and a chain $(v_{20}, v_{21}, v_{22}, v_{23}, v_1, v_2, v_3)$ in the subgraph $(V, \varnothing, E)$. Therefore, a minimal colour of the vertex v_2 is not less than 6; $c(v_2) \geq 6$. However, we see that the minimal colour of the vertex v_2 cannot be equal to 6, 7, 8 and 9, because the following four inclusions hold: $[v_2, v_6] \in E$, $[v_2, v_7] \in E$, $[v_2, v_8] \in E$, $[v_2, v_9] \in E$. There is a path $(v_{24}, v_{25}, v_{26}, v_4, v_5, v_6, v_7, v_8, v_9, v_3)$ in the directed subgraph $(V, A, \varnothing)$ and a chain $(v_{24}, v_{25}, v_{26}, v_4, v_5, v_6, v_7, v_8, v_9, v_3)$ in the subgraph $(V, \varnothing, E)$. Thus, a minimal possible colour of the vertex v_2 is not less than 10, i.e., the inequality $c(v_2) \geq 10$ holds. Hence, the minimal possible colour of the vertex v_3 is not less than 11 because of the inclusions $[v_2, v_3] \in E$ and $(v_2, v_3) \in A$. The inequality $\chi(G) \geq 11$ holds, so the constructed colouring $c(G)$ is optimal. We obtain the optimal semiactive schedule $S = (C(v_1, C(v_2, \dots, C(v_{|W|})$ such that equality $c(v_i) = C(v_i)$ holds for each vertex $v_i \in W$.

7. Minimising the Schedule Length for Interruptible Integer-Time Operations with Integer Release Dates and Equivalent Minimisation of the Maximal Lateness

We next show that finding a makespan-optimal schedule for the general scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with integer release dates and interruptible integer-time operations Q_{ij} is similar to finding the optimal schedule for the problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with integer due dates $D_i \geq 0$, $J_i \in J$, and interruptible integer-time operations Q_{ij}.

Theorem 9. *Every solvable scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with integer release dates $r_i \geq 0$, $J_i \in J$, and interruptible integer-time operations Q_{ij}, $J_i \in J$, is equivalent to the solvable scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with integer due dates $D_i \geq 0$, $J_i \in J$, and interruptible integer-time operations Q_{ij}, $J_i \in J$.*

Proof. The problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ was considered in Theorem 8. The proof of Theorem 8 showed how to construct a mixed graph G that completely determines the input data for a solvable scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$. Since the considered scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ is solvable, this mixed graph G must be colourable according to Theorem 4 and Theorem 8. From Theorem 1 it follows that the directed graph $(V, A, \varnothing)$ does not have a circuit containing adjacent vertices in the graph $(V, \varnothing, E)$.

For the given scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ on the above mixed graph $G = (V, A, E)$, we will show how to construct the corresponding problem

$G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, where due dates $D_i \geq 0$ are integer for all jobs $J_i \in J$, operations Q_{ij} have integer durations for all jobs $J_i \in J$, and interruptions are allowed for all operations.

In the corresponding problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, jobs $J = \left\{J_1, J_2, \ldots, J_{|J|}\right\}$ have a release date of zero $r_i = 0$, $J_i \in J$. Jobs $J = \left\{J_1, J_2, \ldots, J_{|J|}\right\}$ have to be processed by the same set $M = \left\{M_1, M_2, \ldots, M_{|M|}\right\}$ of the dedicated processors. We assume that in the general scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, each processor $M_i \in M$ must process the same set $V_i = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\} \subseteq V$ of the jobs as in the general problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ considered in Sections 5 and 6.

In the corresponding problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, integer due date $D_i \geq 0$ is given for every job $J_i \in J$ as follows: $D_i = \max\{r_j : J_j \in J\} - r_i$.

Let us define a strict order $(v_{k_{|V^{(k)}|}}, v_{k_{|V^{(k)}|-1}}, \ldots, v_{k_2}, v_{k_1})$ for processing the operations $V^{(k)} = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V^{(k)}|}}\right\}$. It is clear that this strict order $(v_{k_{|V^{(k)}|}}, v_{k_{|V^{(k)}|-1}}, \ldots, v_{k_2}, v_{k_1})$ is opposite to the strict order $(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V^{(k)}|}})$ determined for the corresponding problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ for processing integer-time operations $V^{(k)}$.

If the precedence constraints $v_i \Rightarrow v_j$ and the weak precedence constraints $v_p \to v_q$ are defined on the set of operations belonging to different jobs in the corresponding problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$, then the inverse precedence constraint $v_j \Rightarrow v_i$ and the inverse weak precedence constraints $v_q \to v_p$ have to be given for the problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$.

It follows from Theorem 8 that the optimal semiactive schedule for the roblem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ determined by the mixed graph $G = (V, A, E)$ can be defined by the optimal colouring $c(G)$. This colouring $c(G)$ also defines an optimal semiactive schedule for the problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. It is clear that the optimal maximal lateness may be determined as follows: $L_{\max} = \chi(G) - \max\{r_i : J_i \in J\}$. Thus, we proved the following claim.

Every solvable scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with given integer release dates $r_i \geq 0$, $J_i \in J$, and integer durations of interruptible operations Q_{ij} is equivalent to the scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with integer due dates $D_i = \max\{r_j : J_j \in J\} - r_i$, $J_i \in J$, and integer durations of the interruptible operations Q_{ij}, $J_i \in J$.

Obviously, the above arguments are reversible. This allows us to prove the following claim in a similar way.

Every solvable scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with given integer due dates $D_i \geq 0$, $J_i \in J$, and integer durations of the interruptible operations Q_{ij}, is equivalent to the scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ with given integer release dates $r_i = \max\{D_j : J_j \in J\} - D_i$ and integer durations of the interruptible operations Q_{ij}, $J_i \in J$.

Theorem 9 is proved. □

The following corollary follows directly from Theorem 8 and Theorem 9.

Corollary 3. *If the job due dates and durations of operations Q_{ij}, $J_i \in J$, are integers for the solvable scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, then an semiactive optimal schedule for this problem is presentable as an optimal colouring of vertices of the mixed graph G that determines the input data of this scheduling problem.*

To illustrate Theorem 9 and Corollary 3, we solve Example 5 of the problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with four jobs $J = \{J_1, J_2, J_3, J_4\}$, seven dedicated processors $M = \{M_1, M_2, \ldots, M_7\}$, integer due dates $D_i \geq 0$, integer processing times $t_{ij} > 0$ of the multi-processor operations, precedence constraints between operations of different and the same jobs, and allowed interruptions of the integer-time operations.

Example 5. *In Example 5 of the scheduling problem* $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ *on the mixed graph* $G = (V, A, E)$ *depicted in Figure 5, the set of jobs* J_1, J_2, J_3 *and* J_4 *are ready for processing on set* $M = \{M_1, M_2, \ldots, M_9\}$ *of the dedicated processors from time zero* $t = 0$*. Each processor* $M_k \in M$ *must process the same subset of the jobs as in Example 4. However, the orders for fulfilling the same set of operations are opposite to the linear orders determined in Example 4.*

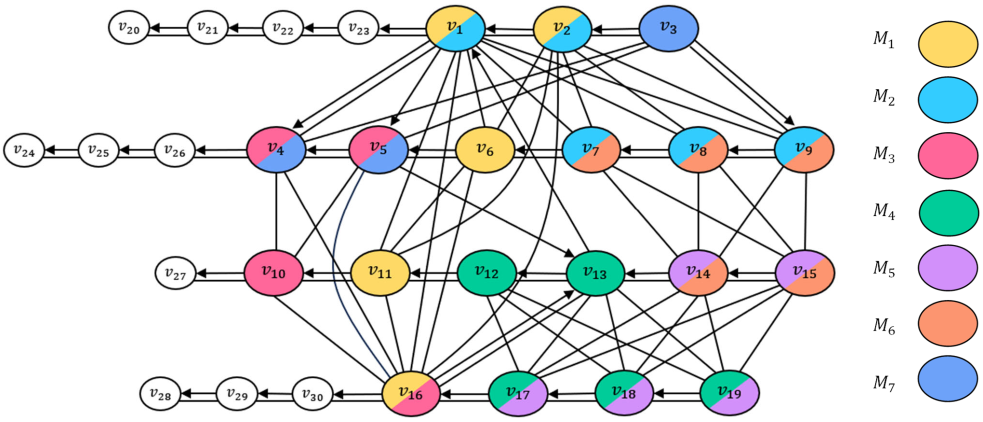

Figure 5. Mixed graph $G^{\leftarrow} = (V, A^{\leftarrow}, E)$ determining Example 5 of the problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$.

While the precedence constraints $v_i \to v_j$ and $v_p \Rightarrow v_q$ are determined between operations different jobs in Example 4, the inverse precedence constraints $v_j \to v_i$ and $v_q \Rightarrow v_p$ are determined in Example 5 of the scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. Thus, in Example 5, the due dates of the given set of jobs $J_i \in J = \{J_1, J_2, J_3, J_4\}$ are defined as follows: $D_1 = \max\{r_j : J_i \in J\} - r_1 = 4 - 4 = 0$, $D_2 = 4 - r_2 = 4 - 3 = 1$, $D_3 = 4 - r_3 = 4 - 1 = 3$, $D_4 = 4 - r_4 = 4 - 3 = 1$.

The input data for Example 5 are determined by the mixed graph $G^{\leftarrow} = (V, A^{\leftarrow}, E)$ depicted in Figure 5. Set V of the vertices and set E of the edges in the mixed graph $G^{\leftarrow}$ are the same as those in the mixed graph G presented in Figure 4.

Compare mixed graph $G^{\leftarrow}$ depicted in Figure 5 for scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ with mixed graph G for scheduling problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ depicted in Figure 4. According to Theorem 9 and Corollary 3, for Example 5 of the scheduling problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ on the mixed graph $G^{\leftarrow} = (V, A^{\leftarrow}, E)$, the optimal semiactive schedule S is determined by the following optimal colouring $c(G^{\leftarrow})$: $c(v_1) = 7$, $c(v_2) = 2$, $c(v_3) = 1$, $c(v_4) = 8$, $c(v_5) = 7$, $c(v_6) = 6$, $c(v_7) = 5$, $c(v_8) = 4$, $c(v_9) = 3$, $c(v_{10}) = 10$, $c(v_{11}) = 9$, $c(v_{12}) = 8$, $c(v_{13}) = 7$, $c(v_{14}) = 6$, $c(v_{15}) = 5$, $c(v_{16}) = 5$, $c(v_{17}) = 4$, $c(v_{18}) = 3$, $c(v_{19}) = 2$.

The colouring $c(G^{\leftarrow})$ determines the optimal semiactive schedule:

$$C(Q_{1,1}) = c(v_3) = 1,\ C(Q_{1,2}) = c(v_1) = 7,\ C(Q_{2,1}) = c(v_7) = 5,\ C(Q_{2,2}) = c(v_6) = 6,$$
$$C(Q_{2,3}) = c(v_4) = 8, C(Q_{3,1}) = c(v_{14}) = 6, C(Q_{3,2}) = c(v_{12}) = 8, C(Q_{3,3}) = c(v_{11}) = 9,$$
$$C(Q_{4,1}) = c(v_{17}) = 4, C(Q_{4,2}) = c(v_{16}) = 5.$$

We compute the optimal objective value as follows:

$$L_{\max} = \chi(G^{\leftarrow}) - \max\{r_i : J_i \in J\} = 11 - 4 = 7.$$

8. Discussion

This research is a continuation of the works [1,2,10–22], where graph colouring models and algorithms were developed for solving various scheduling problems. The developed method is based on graph-theoretic models for scheduling problems. We have studied relationships between optimal colourings of the mixed graphs and general scheduling problems $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ and $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$. In both investigated scheduling problems, it is necessary to construct optimal schedules for fulfilling partially ordered sets of integer-time multi-processor tasks with allowed operation interruptions. Precedence constraints of the two types of completion–start and start–start can be given on the set of integer-time operations. Furthermore, specified subsets of the operations with unit durations must be executed simultaneously.

The propositions proved in Sections 4–7 can be used to construct mixed graphs for representing the entire input data of the numerous scheduling problems with interruptible integer-time operations. It is proven that to construct an optimal semiactive schedule for any studied scheduling problem, it is sufficient to determine the optimal colouring of the vertices of the mixed graph representing all input data of the considered scheduling problem.

It is also shown that notions of graph theory are quite sufficient for the formulation of the problems $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ and $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$, and a huge number of their special cases. To illustrate the obvious benefits of using graph theory terminology instead of terminology of scheduling theory, we list the scheduling terms used in Sections 1–7: *processing system (multi-stage); production planning (scheduling); job-shop, general shop; equipment, machine, processor; schedule (semiactive, feasible, optimal); schedule optimality criterion (makespan, maximal lateness,; production cost); manufacture; raw material; final product; job (simple, first, second, last) task; release date (equal, integer); input data; planning horizon; schedule length; operation (multi-processor, interruptible); (allowed) interruption, preemption; machine route, (first, last) operation; operation duration (unit, integer), operation order (linear, strict); operation assignment to machine; simultaneous execution of operations; fulfill; process; ready for processing; (operation, job, task) start; (operation, job, task) completion; (operation, task) interruption, schedule (makespan, length); three-field notation; capacity constraint; (week) precedence constraints (completion-start, start-start), unit-time operation (task), integer-time (operation, job, task) operation (task) processing time (duration), production conditions (efficiency), due date (integer).*

To obtain and present the same results in notions of mixed graphs, only fewer terms were used, namely: *vertex (incidental, adjacent); arc, edge (multiple); graph (finite, directed, mixed, k-partite, bipartite, tripartite, complete, colourable); digraph; (vertex-induced) subgraph, clique, connected component (graph); colouring (strict, optimal); vertex colour, chromatic number; loop; path, passing; chain (length); circuit.*

In Sections 5 and 7, we proved that every solvable problem of constructing an optimal colouring of the mixed graph reduces to problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ and $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. Thus, for many of the statements proved so far for these scheduling problems and their various special cases, there are similar statements for the problems of defining optimal colourings of the vertices of the mixed graphs.

It is shown that each solvable problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ (as well as the solvable problem $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$) pseudo-polynomially reduces to finding the optimal colouring of a mixed graph. It is worth noting that the pseudo-polynomiality of this reduction is based on the fact that all input data for both scheduling problems are determined by mixed graphs to be coloured for solving these scheduling problems. Therefore, it is unclear and probably impossible to find a polynomial reduction of these scheduling problems to mixed graph colouring problems.

The statements proved for finding optimal colourings of the mixed graphs [6,7,15,17,23–32] could be used for finding a solution of the NP-hard scheduling problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ and $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. To apply the developed mixed graph colouring models, methods, and algorithms, solvable scheduling problems must have positive integer processing times of the multi-processor tasks, non-negative integer release dates of all jobs for the schedule length minimisation (and non-negative integer due dates for the minimisation of the maximal lateness). It should be noted that there is no restriction on the mixed graphs for using the scheduling algorithms for constructing optimal colourings of vertices of the mixed graph.

The results presented and proved in this article can be considered as a substantiation of graph-theoretic methods for solving scheduling problems by reducing them to finding optimal colouring of mixed graphs, which determine the input data of scheduling problems. Conditions and limitations of an individual scheduling problem are determined by the corresponding mixed graph. For the individual scheduling problem, such a mixed graph is defined uniquely.

It should be noted that different scheduling problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ and $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$ can be defined by the same mixed graph. Thus, the results proven for colourings of the particular mixed graph $G = (V, A, E)$ are applicable, as a rule, for a whole set of the scheduling problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ or problems $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. In particular, most results presented and proven in this paper can be used for studying colourings of the specified classes of mixed graphs, which are generated by the scheduling problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ and $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$. This can help attract graph theorists to solve the multistage scheduling problems and their numerous special cases.

9. Conclusions

This paper presents efficient reformulations of a huge number of scheduling problems into mixed graph colouring ones. Sections 1–3 provide an overview of the preliminary results used in the proofs of the new results presented in Sections 4–8. We studied the optimal scheduling of a partially ordered set of multi-processor tasks with allowed interruptions as optimal colourings of the vertices of the corresponding mixed graphs. We proved that the problem of optimal colouring of vertices of an arbitrary mixed graph reduces to the problem of constructing schedules that minimise a maximal lateness for processing partially ordered sets of multi-processor tasks with integer durations, integer due dates and allowed interruptions. We also proved the inverse statements that problems of constructing schedules minimising a maximal lateness for fulfilling partially ordered sets of multi-processor tasks pseudo-polynomially reduce to problems of optimally coloring the vertices of the mixed graphs, which determines all input data of the considered scheduling problems.

The established relationship between the generalisations of scheduling problems and problems of finding an optimal colouring of the vertices of mixed graphs presented in the article makes it possible to solve any solvable problems $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ and $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, as well as numerous special cases of these problems in terms of graph theory, and not to use the special terms of schedule theory, which are related to the practical production planning problems. It turned out that the terms of graph theory are quite enough to formulate any problem $G_cMPT\big|[r_i], [t_{ij}], pmtn\big|C_{\max}$ or $G_cMPT\big|[t_{ij}], pmtn, [D_i]\big|L_{\max}$, and for the development of algorithms for solving problems of schedule theory as a result of the search for optimal coloring of the mixed graphs that determine the input data of scheduling problems.

The presented and generalised theorems and results (Sections 1–3) and new results proved in Sections 4–7 allow making scheduling theory more abstract due to the possibility to use abstract models, effective methods and algorithms for optimal colouring the vertices of the mixed graphs for solving most scheduling problems with bottleneck objectives. Such investigations of the relationships between optimisation problems of scheduling theory and those of graph theory may be continued further.

Future research directions may be oriented towards developing mixed graph colouring models and algorithms suitable for solving scheduling problems with other regular optimality criteria. In particular, non-bottleneck criteria are widely used in scheduling theory [5,6,9,22–26]. It should be noted that both criteria considered in this paper are bottleneck criteria. We plan to study other scheduling problems that are reducible or equivalent to colouring the vertices of the corresponding mixed graphs. A promising research may be aimed to usage of the mixed graph models for developing efficient algorithms for scheduling the cloud computations. One could assume that durations of all tasks are integer and operation preemptions are allowed. Both studied objectives (schedule length and maximal lateness) may be considered for cloud computations.

We plan to study other classes of the multistage scheduling problems that are equivalent to problems of optimal colorings of vertices of the mixed graphs. We plan also to consider polynomially solvable multistage scheduling problems, which will allow us to discover polynomially solvable cases of the problem of finding optimal colourings $c(G)$ of the specific mixed graphs $G = (V, A, E)$.

Author Contributions: Conceptualisation, Y.N.S.; methodology, Y.N.S.; validation, E.I.M.; formal analysis, E.I.M.; investigation, E.I.M. and Y.N.S.; data curation, E.I.M.; writing—original draft, Y.N.S.; writing—review and editing, Y.N.S.; visualisation, E.I.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research of Y.S. was funded by Belarusian Republican Foundation for Fundamental Research, grant number Φ23PHΦ-017.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Example A1. *The individual problem* $G_c|t_{ij} = 1|C_{\max}$ *determined by mixed graph* $G^* = (V, A^*, E^*)$ *is analogous to Example 1 of a job-shop unit-time problem* $J|t_{ij} = 1|C_{\max}$*, with the exception that there are given start-start constraints and completion-start constraints on the set of operations of different jobs of set* J*. All the additional constraints are determined as follows:* $v_1 \Rightarrow v_4$*;* $v_4 \Rightarrow v_3$*;* $v_3 \to v_1$*;* $v_8 \to v_4$ *and* $v_{12} \to v_9$*. Thus, the appropriate mixed graph* $G^* = (V, A^*, E^*)$ *has to contain an additional subset of arcs and subset set of edges defined as follows:*

$$A^* = A \cup \{(v_1, v_4), (v_4, v_3), (v_3, v_1), (v_8, v_4), (v_{12}, v_9)\};\ E^* = E \cup \{[v_8, v_4], [v_{12}, v_9]\}.$$

Note that the operations v_1, v_3 and v_4 have to be fulfilled simultaneously in ever schedule since circuit (v_1, v_4, v_3, v_1) exists in digraph $(V, \varnothing, E^*)$, which is a subgraph of the mixed graph $G^* = (V, A^*, E^*)$ dipicted in Figure A1. At the right side of Figure A1, the colors of coloring $c(G^*)$ are presented instead of the natural numbers: $c(v_1) = 2$, $c(v_2) = 1$, $c(v_3) = 2$, $c(v_4) = 2$, $c(v_5) = 3$, $c(v_6) = 4$, $c(v_7) = 5$, $c(v_8) = 1$, $c(v_9) = 3$, $c(v_{10}) = 5$, $c(v_{11}) = 1$, $c(v_{12}) = 2$. The colouring $c(G^*)$ is optimal, $\chi(G^*) = 5$, since directed graph $(V, A^*, \varnothing)$ contains path $(v_8, v_4, v_5, v_6, v_7)$ and graph $(V, \varnothing, E^*)$ contains chain $(v_8, v_4, v_5, v_6, v_7)$, whose lengths are equal to 5.

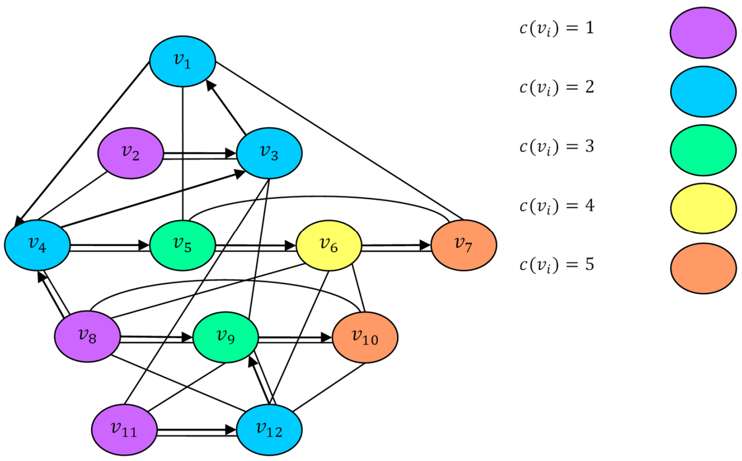

Figure A1. Mixed graph $G^* = (V, A^*, E^*)$ that determines Example A1 of the scheduling problem $G_c|t_{ij} = 1|C_{\max}$.

Due to these path and chain, the following inequality holds: $\chi(G^*) \geq 5$. From Theorem 3, it follows that the optimal colouring $c(G^*)$ defines the following schedule S that is optimal for Example A1 of the unit-time problem $G_c|t_{ij} = 1|C_{\max}$, where the input data are determined by mixed graph G^* depicted in Figure A1.

$$S = (C(Q_{1,1}) = 2, C(Q_{2,1}) = 1, (Q_{2,2}) = 2, C(Q_{3,1}) = 2, C(Q_{3,2}) = 3, C(Q_{3,3}) = 4, C(Q_{3,4}) = 5,$$
$$C(Q_{4,1}) = 1, C(Q_{4,2}) = 3, C(Q_{4,3}) = 5, C(Q_{5,1}) = 1, C(Q_{5,2}) = 2)$$

The correspondence of the operations $Q_{ij} \in Q$ and the vertices $v_k \in V$ is determined based on Remark 1. The equalities $\chi(G^*) = 5 = C_{\max}$ hold due to Theorem 3.

Appendix B

Example A2. *In Example A2, the set $J = \{J_1, J_2, J_3, J_4, J_5\}$ of jobs is the same as in Example 1 and Example A1. The set of processors is increased from four processors in Examples 1 and Example A1 to nine processors in Example A2: $M = \{M_1, M_2, \dots, M_9\}$.*

The processor M_1 processes the operations $V_1 = \{v_1, v_7\}$. In Figure A2, the forbiddance to process the operations from the set V_1 simultaneously on the processor M_1 is determined by the edge $[v_1, v_7]$ in the graph $(V, \varnothing, E)$. The processor M_2 processes the operations $V_2 = \{v_2, v_4\}$. The forbiddance to process the operations from the set V_2 simultaneously on the processor M_2 is determined by the edge $[v_2, v_4]$.

Similarly, the forbiddance to process operations from the set $V_3 = \{v_3, v_9\}$ simultaneously on the processor M_3 is determined by the edge $[v_3, v_9]$. The forbiddance to process this pair of operations from the set $V_4 = \{v_6, v_{12}\}$ simultaneously on the processor M_4 is determined by the edge $[v_6, v_{12}]$ in the graph $(V, \varnothing, E)$. The forbiddance to process operations from the set $V_5 = \{v_5, v_{11}\}$ simultaneously on the processor M_5 is determined by the edge $[v_5, v_{11}]$. The forbiddance to process the operations from the set $V_6 = \{v_8, v_{12}\}$ simultaneously on the processor M_6 is determined by the edge $[v_8, v_{12}]$. The forbiddance to process the operations from the set $V_7 = \{v_7, v_{10}\}$ simultaneously on the processor M_7 is determined by the edge $[v_7, v_{10}]$. The forbiddance to process the operations from the set $V_8 = \{v_9, v_{11}\}$ simultaneously is determined by the edge $[v_9, v_{11}]$. The forbiddance to process the operations from the set $V_9 = \{v_3, v_4\}$ simultaneously on processor M_9 is determined by the edge $[v_3, v_4]$. The processors, which are used to process the operation $v_i \in V$, are presented near the vertex v_i in Figure A2. The precedence constraints established between operations of different jobs are given as follows: $v_1 \Rightarrow v_4$; $v_1 \Rightarrow v_8$; $v_3 \Rightarrow v_1$; $v_8 \Rightarrow v_3$; $v_6 \Rightarrow v_{10}$; $v_{10} \Rightarrow v_{12}$; $v_8 \to v_4$; $v_{11} \to v_8$. These precedence constraints are represented by seven arcs $\{(v_1, v_4), (v_1, v_8), (v_3, v_1), (v_8, v_3), (v_6, v_{10}), (v_{10}, v_{12}), (v_8, v_4)\} \subset A$ and two edges $\{[v_4, v_8], [v_8, v_{11}]\} \subset E$.

The digraph $(V, A, \varnothing)$ has one circuit (v_1, v_8, v_3, v_1) and this circuit has no adjacent vertices in the graph $(V, \varnothing, E)$. Since the digraph $(V, A, \varnothing)$ has no circuit with adjacent vertices in the graph $(V, \varnothing, E)$, there exists a schedule for Example A2 of the general unit-time problem $G_cMPT|t_i = 1|C_{\max}$ that is completely determined by the mixed graph G; see Theorem 4. We construct the following colouring $c(G)$ of the mixed graph $G = (V, A, E)$: $c(v_1) = 2$, $c(v_2) = 1$, $c(v_3) = 2$, $c(v_4) = 3$, $c(v_5) = 4$, $c(v_6) = 5$, $c(v_7) = 6$, $c(v_8) = 2$, $c(v_9) = 3$, $c(v_{10}) = 5$, $c(v_{11}) = 1$, $c(v_{12}) = 6$. The colouring $c(G)$ is optimal, $\chi(G) = 6$, since the directed graph $(V, A, \varnothing)$ contains the path $(v_{11}, v_8, v_4, v_5, v_6, v_7)$ and the subgraph $(V, \varnothing, E)$ of the mixed graph G contains the chain $(v_{11}, v_8, v_4, v_5, v_6, v_7)$, whose lengths are equal to 6. Because of this path and chain, the inequality $\chi(G) \geq 6$ holds. By Theorem 5, the optimal colouring $c(G)$ defines the optimal semiactive schedule (1) for Example A2 of the unit-time problem $G_c|t_{ij} = 1|C_{\max}$, whose input data is determined by the mixed graph G depicted in Figure A2. Based on Remark 1 about the correspondence of operations $Q_{ij} \in Q$ and vertices $v_k \in V$, we obtain the following makespan-optimal semiactive schedule:

$$S = (C(Q_{1,1}) = 2, C(Q_{2,1}) = 1, (Q_{2,2}) = 2, C(Q_{3,1}) = 3, C(Q_{3,2}) = 4, C(Q_{3,3}) = 5, C(Q_{3,4}) = 6,$$
$$C(Q_{4,1}) = 2, C(Q_{4,2}) = 3, C(Q_{4,3}) = 5, C(Q_{5,1}) = 1, C(Q_{5,2}) = 6)$$

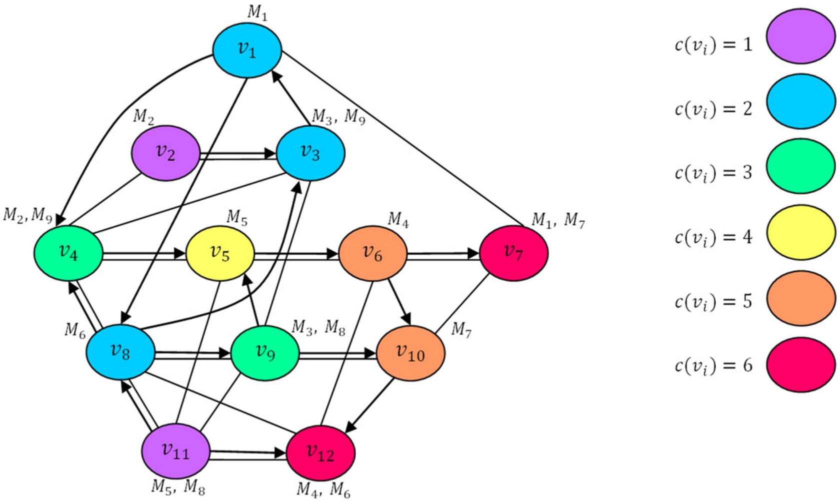

Figure A2. Mixed graph $G = (V, A, E)$ that defines the input data for Example A2.

By Theorem 5, the equality $\chi(G) = 6 = C_{\max}(S)$ must hold.

Note that Theorem 5 is only valid for colourable mixed graphs. For example, if the precedence constraint $v_{12} \Rightarrow v_9$ is added to the input data of Example A2, we obtain the following mixed graph: $G^+ = (V, A \cup \{(v_{12}, v_9)\}, E)$. The obtained mixed graph becomes un-colourable due to the existence of a circuit $(v_{12}, v_9, v_{10}, v_{12})$ such that the vertices v_9 and v_{10} are adjacent in the graph $(V, \varnothing, E)$, which is a subgraph of the mixed graph $G^+ = (V, A \cup \{(v_{12}, v_9)\}, E)$; see Theorem 1. Therefore, the modified Example A2 (with the additional precedence constraint $v_{12} \Rightarrow v_9$) has no feasible schedule due to Theorem 4.

References

1. Sotskov, Y.N.; Dolgui, A.; Werner, F. Mixed graph coloring for unit-time job-shop scheduling. *Int. J. Math. Algorithms* **2001**, *2*, 289–323.
2. Sotskov, Y.N.; Tanaev, V.S.; Werner, F. Scheduling problems and mixed graph colorings. *Optimization* **2002**, *51*, 597–624.
3. Sotskov, Y.N.; Tanaev, V.S. A chromatic polynomial of a mixed graph. *Vestsi Akad. Navuk BSSR Ser. Fiz. Mat. Navuk* **1976**, *6*, 20–23. (In Russian)
4. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Miller, R.E., Thatcher, J.W., Eds.; Plenum Press: New York, NY, USA, 1972; pp. 85–103.
5. Brucker, P. *Scheduling Algorithms*; Springer: Berlin, Germany, 1995.
6. Harary, F. *Graph Theory*; Reading; Addison-Wesley: Boston, MA, USA, 1969.
7. Thulasiraman, K.; Swamy, M.N.S. *Graphs: Theory and Algorithms*; John Wiley & Sons, Inc.: Toronto, ON, Canada, 1992.
8. Hansen, P.; Kuplinsky, J.; de Werra, D. Mixed graph colorings. *Math. Methods Oper. Res.* **1997**, *45*, 145–160. [CrossRef]
9. Sotskov, Y.N. Mixed graph colouring as scheduling multi-processor tasks with equal processing times. *J. Belarusian State Univ. Math. Inform.* **2021**, *2*, 67–81. [CrossRef]
10. Giaro, K.; Kubale, M.; Obszarski, P. A graph coloring approach to scheduling of multiprocessor tasks on dedicated machines with availability constraints. *Discret. Appl. Math.* **2009**, *157*, 3625–3630. [CrossRef]
11. Hoogeveen, J.A.; Lenstra, J.K.; Veltman, B. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *Eur. J. Oper. Res.* **1996**, *89*, 172–175. [CrossRef]
12. Sotskov, Y.N.; Mihova, E.I. Scheduling multiprocessor tasks with equal processing times as a mixed graph coloring problem. *Algorithms* **2021**, *14*, 246. [CrossRef]
13. De Werra, D. Restricted coloring models for timetabling. *Discret. Math.* **1997**, *165/166*, 161–170. [CrossRef]
14. De Werra, D. On a multiconstrained model for chromatic scheduling. *Discret. Appl. Math.* **1999**, *94*, 171–180. [CrossRef]
15. Damaschke, P. Parameterized mixed graph coloring. *J. Comb. Optim.* **2019**, *38*, 326–374. [CrossRef]
16. Kouider, A.; Ait Haddadne, H.; Ourari, S.; Oulamara, A. Mixed graph coloring for unit-time scheduling. *Int. J. Prod. Res.* **2017**, *55*, 1720–1729. [CrossRef]
17. Kouider, A.; Ait Haddadene, H.; Oulamara, A. On minimization of memory usage in branch-and-bound algorithm for the mixed graph coloring: Application to the unit-time job shop scheduling. *Comput. Oper. Res.* **2019**, *4967*, 1001–1008.
18. Kouider, A.; Ait Haddadne, H. A bi-objective branch-and-bound algorithm for the unit-time job shop scheduling: A mixed graph coloring approach. *Comput. Oper. Res.* **2021**, *132*, 105319. [CrossRef]

19. Baaziz, A.; Haddadene, H.A.; Oulamara, A.; Kouider, A. Scheduling preemptive jobs on parallel machines with a conflict graph: A graph multi-colouring approach. *Int. J. Math. Oper. Res.* **2023**, *25*, 47–67. [CrossRef]
20. Beck, M.; Blado, D.; Crawford, J.; Jean-Louis, T.; Young, M. On weak chromatic polynomials of mixed graphs. *Graphs Comb.* **2015**, *31*, 91–98. [CrossRef]
21. Niu, D.; Liu, B.; Zhang, H.; Yin, M. Improving local search for the weighted sum coloring problem using the branch-and-bound algorithm. *Knowl. Based Syst.* **2022**, *246*, 108703. [CrossRef]
22. Jacqueline, O.; Alden, M.; Golshan, M.; Seyedamirabbas, M. Graph-based modeling in shop scheduling problems: Review and extensions. *Appl. Sci.* **2021**, *11*, 4741. [CrossRef]
23. Liang, P.; Fu, Y.; Gao, K. Multi-product disassembly line balancing optimization method for high disassembly profit and low energy consumption with noise pollution constraints. *Eng. Appl. Artif. Intell.* **2024**, *130*, 107721. [CrossRef]
24. Fu, Y.; Zhou, M.; Guo, X.; Qi, L.; Gao, K.; Albeshri, A. A multiobjective scheduling of energy-efficient stochastic hybrid open shop with brain storm optimization and simulation evaluation. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *54*, 4260–4272. [CrossRef]
25. Fu, Y.; Ma, X.; Gao, K.; Li, Z.; Dong, Y. Multi-objective home health care routing and scheduling with sharing service via a problem-specific knowledge-based artificial bee colony algorithm. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 1706–1719. [CrossRef]
26. Ma, X.; Fu, Y.; Gao, K.; Zhu, L.; Sadollah, A. A multi-objective scheduling and routing problem for home health care services via brain storm optimization. *Complex Syst. Model. Simul.* **2023**, *3*, 32–46. [CrossRef]
27. Andrade, E.; Bonifácio, A.E.; Robbiano, M.; Rodríguez, J.; Tapia, K. Some families of integral mixed graphs. *Linear Algebra Appl.* **2022**, *641*, 48–66. [CrossRef]
28. Mostafaie, T.; Modarres Khiyabani, F.; Navimipour, N.J. A systematic study on meta-heuristic approaches for solving the graph coloring problem. *Comput. Oper. Res.* **2020**, *120*, 104850. [CrossRef]
29. Balakrishnan, S.; Suresh, T.; Marappan, R.; Venkatesan, R.; Sabri, A. New hybrid decentralized evolutionary approach for DIMACS challenge graph coloring & wireless network instances. *Int. J. Cogn. Comput. Eng.* **2023**, *4*, 259–265.
30. Astuti, W.; Adiwijaya. Graph coloring based on evolutionary algorithms to support data hiding scheme on medical images. *Procedia Comput. Sci.* **2015**, *74*, 173–177. [CrossRef]
31. Agrawal, J.; Agrawal, S. Acceleration based particle swarm optimization for graph coloring problem. *Procedia Comput. Sci.* **2015**, *60*, 714–721. [CrossRef]
32. Assi, M.; Halawi, B.; Haraty, R.A. Genetic algorithm analysis using the graph coloring method for solving the university timetable problem. *Procedia Comput. Sci.* **2018**, *126*, 899–906. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.