УДК 681.32

РАСКРАСКА СМЕШАННОГО ГРАФА КАК ПОСТРОЕНИЕ РАСПИСАНИЯ ОБСЛУЖИВАНИЯ МНОГОПРОЦЕССОРНЫХ ТРЕБОВАНИЙ С ОДИНАКОВЫМИ ДЛИТЕЛЬНОСТЯМИ

Ю. Н. СОТСКОВ[1)]

[1)]*Объединенный институт проблем информатики НАН Беларуси, ул. Сурганова, 6, 220012, г. Минск, Беларусь*

Задача обслуживания частично упорядоченных единичных требований последовательными приборами формулируется как раскраска смешанного графа, т. е. как назначение целых чисел (цветов) $\{1, 2, \ldots, t\}$ вершинам (требованиям) $V = \{v_1, v_2, \ldots, v_n\}$ смешанного графа $G = (V, A, E)$, при котором вершины v_p и v_q, инцидентные ребру $[v_p, v_q] \in E$, имеют различные цвета. А при наличии дуги $(v_i, v_j) \in A$ цвет вершины v_i не превосходит цвет вершины v_j. Доказано, что оптимальная раскраска смешанного графа $G = (V, A, E)$ эквивалентна задаче $GcMPT|p_i = 1|C_{\max}$ поиска оптимального расписания обслуживания частично упорядоченных требований с единичными (одинаковыми) длительностями. В отличие от классических задач построения расписаний в рассматриваемой задаче $GcMPT|p_i = 1|C_{\max}$ необходимо несколько различных приборов для обслуживания отдельного требования. Помимо отношений предшествования, заданных на множестве требований $V = \{v_1, v_2, \ldots, v_n\}$,

Образец цитирования:
Сотсков ЮН. Раскраска смешанного графа как построение расписания обслуживания многопроцессорных требований с одинаковыми длительностями. *Журнал Белорусского государственного университета. Математика. Информатика.* 2021;2:67–81 (на англ.).
https://doi.org/10.33581/2520-6508-2021-2-67-81

For citation:
Sotskov YuN. Mixed graph colouring as scheduling multi-processor tasks with equal processing times. *Journal of the Belarusian State University. Mathematics and Informatics.* 2021;2:67–81.
https://doi.org/10.33581/2520-6508-2021-2-67-81

Автор:
Юрий Назарович Сотсков – доктор физико-математических наук, профессор; главный научный сотрудник лаборатории математической кибернетики.

Author:
Yuri N. Sotskov, doctor of science (physics and mathematics), full professor; chief researcher at the laboratory of mathematical cybernetics.
sotskov48@mail.ru
https://orcid.org/0000-0002-9971-6169

должно выполняться некоторое подмножество требований одновременно. На основании доказанных в статье теорем утверждается, что множество аналитических результатов, полученных ранее для задач $GcMPT|p_i=1|C_{\max}$, имеют аналоги для оптимальных раскрасок смешанных графов $G=(V,\ A,\ E)$, и наоборот.

Ключевые слова: оптимизация; расписание с единичными длительностями; быстродействие; смешанный граф; вершинная раскраска.

Благодарность. Это исследование выполнено при частичной финансовой поддержке Белорусского республиканского фонда фундаментальных исследований (проект № Ф21-010).

MIXED GRAPH COLOURING AS SCHEDULING MULTI-PROCESSOR TASKS WITH EQUAL PROCESSING TIMES

Yu. N. SOTSKOV[a]

[a]*United Institute of Informatics Problems, National Academy of Sciences of Belarus,*
6 Surhanava Street, Minsk 220012, Belarus

A problem of scheduling partially ordered unit-time tasks processed on dedicated machines is formulated as a mixed graph colouring problem, i. e., as an assignment of integers (colours) $\{1,\ 2,\ \ldots,\ t\}$ to the vertices (tasks) $V=\{v_1,\ v_2,\ \ldots,\ v_n\}$ of the mixed graph $G=(V,\ A,\ E)$ such that if vertices v_p and v_q are joined by an edge $[v_p,\ v_q]\in E$, their colours have to be different. Further, if two vertices v_i and v_j are joined by an arc $(v_i,\ v_j)\in A$, the colour of vertex v_i has to be no greater than the colour of vertex v_j. We prove that an optimal colouring of a mixed graph $G=(V,\ A,\ E)$ is equivalent to the scheduling problem $GcMPT|p_i=1|C_{\max}$ of finding an optimal schedule for partially ordered multi-processor tasks with unit (equal) processing times. Contrary to classical shop-scheduling problems, several dedicated machines are required to process an individual task in the scheduling problem $GcMPT|p_i=1|C_{\max}$. Moreover, along with precedence constraints given on the set $V=\{v_1,\ v_2,\ \ldots,\ v_n\}$, it is required that a subset of tasks must be processed simultaneously. Due to the theorems proved in this article, most analytical results that have been proved for the scheduling problems $GcMPT|p_i=1|C_{\max}$ so far, have analogous results for optimal colourings of the mixed graphs $G=(V,\ A,\ E)$, and vice versa.

Keywords: optimisation; unit-time scheduling; makespan; mixed graph; vertex colouring.

Acknowledgements. This work was partially supported by the Belarusian Republican Foundation for Fundamental Research (project No. Ф21-010).

Introduction

Scheduling models with the prerequisite of equal (or unit) processing times to all given tasks are an approximation of coping with mass-industrial productions and manufacturing of similar items, particularly for a job-shop manufacturing problem that allows managers to personalise each individual item [1]. Such a scheduling problem with unit-times and the minimisation of the makespan is equivalent to an optimal graph colouring that consists of assigning a minimal number of colours to vertices of the graph such that no two adjacent vertices have the same colour. When a scheduling problem requires both precedence and incompatibility constraints, one needs to use a mixed graph colouring introduced in [2] for a formulation of the unit-time scheduling problem. Since the publication of article [2] in 1976, many studies of unit-time scheduling problems with the makespan criterion are based on mixed graph colourings.

Let $G=(V,\ A,\ E)$ denote a finite mixed graph with non-empty set $V=\{v_1,\ v_2,\ \ldots,\ v_n\}$ of the vertices placed at the first position in parenthesis, arc set A at the second position, and edge set E at the third position. An arc $(v_i,\ v_j)\in A$ defines the ordered pair of vertices v_i and v_j. An edge $[v_p,\ v_q]\in E$ means an unordered pair of vertices v_p and v_q. In what follows, we assume that a mixed graph $G=(V,\ A,\ E)$ contains no multiple arcs, no multiple edges, and no loops. If the set A is empty, we have a graph $G=(V,\ \varnothing,\ E)$. If the set E is empty, we have a digraph $G=(V,\ A,\ \varnothing)$. In article [2], a mixed graph colouring is introduced as follows.

Definition 1 [2]. An integer-valued function $c:V\to\{1,\ 2,\ \ldots,\ t\}$ is a colouring $c(G)$ of the mixed graph $G=(V,\ A,\ E)$, if the non-strict inequality $c(v_i)\le c(v_j)$ holds for each arc $(v_i,\ v_j)\in A$ and $c(v_p)\ne c(v_q)$ for each edge $[v_p, v_q] \in E$. A mixed graph colouring $c(G)$ is optimal, if it uses a minimal possible number $\chi(G)$ of different colours $c(v_i) \in \{1, 2, \ldots, t\}$, such a minimal number $\chi(G)$ being called a chromatic number of the mixed graph $G = (V, A, E)$.

If $A = \varnothing$, a colouring $c(G)$ is the usual colouring of the vertices of the graph $G = (V, \varnothing, E)$. Contrary to a colouring of the vertices of the graph $G = (V, \varnothing, E)$ existing for any graph $G = (V, \varnothing, E)$, a mixed graph $G = (V, A, E)$ with $A \neq \varnothing$ and $E \neq \varnothing$ may be uncolourable. A criterion for the existence of a colouring $c(G)$ for the mixed graph G is proved in [2].

Theorem 1 [2]. *A colouring $c(G)$ of the mixed graph $G = (V, A, E)$ exists if and only if the digraph $(V, A, \varnothing)$ has no circuit containing adjacent vertices in the graph $(V, \varnothing, E)$.*

A mixed graph $G = (V, A, E)$ is colourable, if there exists a colouring $c(G)$ of the mixed graph G, otherwise, a mixed graph $G = (V, A, E)$ is uncolourable.

Finding an optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ is an NP-hard problem, even if $A = \varnothing$ [3]. In articles [4; 5], it is shown that a job-shop scheduling problem with unit processing times of all operations and the minimisation of a schedule length (makespan) may be represented as an optimal colouring $c(G)$ of the specified mixed graph $G = (V, A, E)$. In article [6], it is shown that any job-shop scheduling problem with unit processing times of all operations and the minimisation of a total completion time (TCT) may be represented as a mixed graph colouring $c(G)$ minimising a sum of colours of path-endpoints of the specified mixed graph $G = (V, A, E)$ (see also articles [7; 8]).

The unit-time scheduling problem with minimising makespan is NP-hard even for three dedicated machines (processors) [9]. The complexity of a job-shop scheduling problem with a fixed number of jobs (and a fixed number of machines) is investigated in articles [10–13].

Since the NP-hard unit-time flow-shop scheduling problem [14] is polynomially reduced to the job-shop scheduling problem to minimise the TCT, the latter problem is also NP-hard. The complexity of a job-shop scheduling problem with any regular criterion is investigated in [10; 11; 13; 15]. The complexity of a mixed shop-scheduling problem is studied in [16; 17]. A different connection between mixed graph colourings and unit-time shop-scheduling problems is studied in [18–24]. Article [25] presents a comprehensive survey on mixed graph colourings and the equivalent unit-time shop-scheduling problems.

In our article, we show that an optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ is equivalent to finding an optimal schedule for partially ordered multi-processor tasks with unit processing times (or with equal processing times). Contrary to a classical shop-scheduling problem, several dedicated machines are used simultaneously by a task during the complete processing period. Along with the precedence constraints, which are given on the set $V = \{v_1, v_2, \ldots, v_n\}$ of multi-processor tasks, it is required that a subset of tasks must be processed simultaneously. Due to the proven equivalence of the above scheduling problem and the mixed graph colouring $c(G)$, most claims that have been proved so far for a wide class of scheduling problems (without operation preemptions) have analogous claims for optimal mixed graph colourings $c(G)$, and vice versa. Throughout this article, we use the terminology from [26; 27] for graph theory and that from [28; 29] for scheduling theory.

Two classes of shop-scheduling problems as mixed graph colourings

To classify shop-scheduling problems, one can use a three-field notation $\alpha|\beta|\gamma$ introduced in [30], where α specifies a task system and machine environments, β is job characteristics, and γ is an objective function (see [29] for the extensions of classifying parameters).

General shop-scheduling problems with unit-time tasks and minimising makespan. In the general shop unit-time minimum-length scheduling problem denoted by $G|t_i = 1|C_{\max}$, a job set $J = \{J_1, J_2, \ldots, J_{|J|}\}$ must be optimally processed on the different (i. e., dedicated) machines $M = \{M_1, M_2, \ldots, M_{|M|}\}$. We next describe the scheduling problem $G|t_i = 1|C_{\max}$ along with our presentation of this problem by means of the mixed graph colouring $c(G)$.

In the problem $G|t_i = 1|C_{\max}$, a job $J_k \in J$ consists of a set $V^{(k)}$ of linearly ordered operations. The processing time t_i of each operation v_i in the set $V = \bigcup_{k=1}^{|J|} V^{(k)}$ is equal to 1; $t_i = 1$. Due to definition 1, we present every job $J_k \in J$ as a union of path $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{r_k}}\right)$ in the directed subgraph $(V, A, \varnothing)$ and the chain $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{r_k}}\right)$ in subgraph $(V, \varnothing, E)$ of the mixed graph $G = (V, A, E)$, determining input data for the problem $G|t_i = 1|C_{\max}$. As a result, we define a vertex set $V = \bigcup_{k=1}^{|J|} V^{(k)}$ of the mixed graph $G = (V, A, E)$, a subset $E^* = \bigcup_{k=1}^{|J|} \left\{\left[v_{k_1}, v_{k_2}\right], \left[v_{k_2}, v_{k_3}\right], \ldots, \left[v_{k_{r_k}-1}, v_{k_{r_k}}\right]\right\}$ of the edge set $E \supseteq E^*$, and a subset A^* of the arc set A determined by the following implication:

$$\left[v_i, v_j\right] \in E^* \Rightarrow \left(v_i, v_j\right) \in A^*. \tag{1}$$

In the general shop-scheduling problem $G|t_i = 1|C_{\max}$, along with a linear order given on the set $V^{(k)}$ of all operations belonging to the same job $J_k \in J$, there are also given the precedence relations between operations belonging to different jobs in the set J. Let $A \backslash A^*$ denote a subset of set A such that implication (1) does not hold for each arc $\left(v_i, v_j\right) \in A \backslash A^*$. All the given precedence relations make up the precedence constraints.

In the problem $G|t_i = 1|C_{\max}$, a specified machine from the set $M = \left\{M_1, M_2, \ldots, M_{|M|}\right\}$ is required to process operation v_i from the set $V = \bigcup_{k=1}^{|J|} V^{(k)}$. Let $V_i = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\} \subseteq V$ denote a set of all operations processed on machine $M_i \in M$. Any pair of operations requiring the same machine $M_i \in M$ cannot be processed simultaneously [28; 29; 31–33]. We represent all such incompatibility constraints for processing operations $V_i \subseteq V$ on machine $M_i \in M$ (called capacity constraints) by cliques $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ in the subgraph $\left(V, \varnothing, E \backslash E^*\right)$ of the mixed graph $G = (V, A, E)$ constructed for the problem $G|t_i = 1|C_{\max}$. The general shop-scheduling problem $G|t_i = 1|C_{\max}$ is to find a schedule for processing partially ordered operations $V = \bigcup_{i=1}^{|M|} V_i = \bigcup_{k=1}^{|J|} V^{(k)}$, whose length (makespan) $C_{\max} = \max\left\{C_1, C_2, \ldots, C_{|J|}\right\}$ is minimised among lengths of all feasible schedules. Hereafter, C_k denotes a completion time of the job $J_k \in J$. The minimisation of schedule length $C_{\max}$ for partially ordered operations V with unit processing times is reduced to the optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$, where the vertex set V is a set of operations, the arc set A determines the precedence constraints, and the edge set E determines the capacity constraints. More precisely, the union $A^* \bigcup E^*$ of the arc set A^* and the edge set E^* determines $|J|$ subsets $V^{(k)}$ of linearly ordered operations of the jobs $J_k \in J$. The subset $E \backslash E^*$ of edges determines $|M|$ cliques $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ in the graph $\left(V, \varnothing, E \backslash E^*\right)$, where all operations $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ are processed on machine $M_i \in M$. The precedence relations between operations belonging to different jobs are determined in the directed subgraph $\left(V, A \backslash A^*, \varnothing\right)$ of the mixed graph $G = (V, A, E)$.

To illustrate the above reduction of the problem $G|t_i = 1|C_{\max}$ to the optimal colouring $c(G)$, we consider example 1 of the problem $G|t_i = 1|C_{\max}$ with four jobs and six machines (fig. 1). Let the machine set $M = \left\{M_1, M_2, \ldots, M_6\right\}$ have to process the job set $J = \left\{J_1, J_2, J_3, J_4\right\}$. Job $J_1 \in J$ consists of the set $V^{(1)} = \left\{v_1, v_2, v_3\right\}$ of linearly ordered operations. Job $J_1 \in J$ is represented by a union of the path $\left(v_1, v_2, v_3\right)$ in the digraph $(V, A, \varnothing)$ and the chain $\left(v_1, v_2, v_3\right)$ in the graph $(V, \varnothing, E)$. Job $J_2 \in J$ consists of the set $V^{(2)} = \left\{v_4, v_5, v_6, v_7, v_8\right\}$ of linearly ordered operations. Job $J_2 \in J$ is represented by a union of the path $\left(v_4, v_5, v_6, v_7, v_8\right)$ in the digraph $(V, A, \varnothing)$ and the chain $\left(v_4, v_5, v_6, v_7, v_8\right)$ in the graph $(V, \varnothing, E)$. Job $J_3 \in J$ consists of the set $V^{(3)} = \left\{v_9, v_{10}, v_{11}, v_{12}\right\}$ of linearly ordered operations. Job $J_3 \in J$ is represented by a union of the path $\left(v_9, v_{10}, v_{11}, v_{12}\right)$ in the digraph $(V, A, \varnothing)$ and the chain $\left(v_9, v_{10}, v_{11}, v_{12}\right)$ in the graph $(V, \varnothing, E)$. Job $J_4 \in J$ consists of the set $V^{(4)} = \left\{v_{13}, v_{14}, v_{15}\right\}$ of linearly ordered operations. Job $J_4 \in J$ is represented by a union of the path $\left(v_{13}, v_{14}, v_{15}\right)$ in the digraph $(V, A, \varnothing)$ and the chain $\left(v_{13}, v_{14}, v_{15}\right)$ in the graph $(V, \varnothing, E)$.

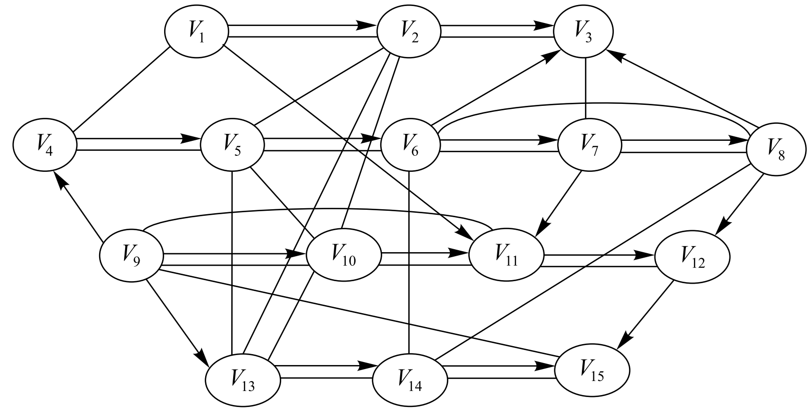

Fig. 1. Mixed graph $G = (V, A, E)$ determining example 1 of the problem $G|t_i = 1|C_{\max}$ with four jobs and six machines, the optimal mixed graph colouring $c(G)$ being equivalent to example 1

Machine M_1 processes operations of the set $V_1 = \{v_1, v_4\}$. The forbiddance to process operations from set V_1 simultaneously is represented by the clique $\{v_1, v_4\}$ in graph $(V, \varnothing, E)$. Machine M_2 processes operations $V_2 = \{v_2, v_5, v_{10}, v_{13}\}$. The forbiddance to process each pair of operations from set V_2 simultaneously is represented by the clique $\{v_2, v_5, v_{10}, v_{13}\}$ in graph $(V, \varnothing, E)$. Machine M_3 processes operations $V_3 = \{v_3, v_7\}$. The forbiddance to process operations from set V_2 simultaneously is represented by the clique $\{v_3, v_7\}$ in graph $(V, \varnothing, E)$. Machine M_4 processes operations $V_3 = \{v_9, v_{11}, v_{15}\}$. The forbiddance to process each pair of operations from set V_2 simultaneously is represented by the clique $\{v_9, v_{11}, v_{15}\}$ in graph $(V, \varnothing, E)$. Machine M_5 processes operations $V_5 = \{v_6, v_8, v_{14}\}$. The forbiddance to process each pair of operations from set V_2 simultaneously is represented by the clique $\{v_6, v_8, v_{14}\}$ in graph $(V, \varnothing, E)$. Machine M_6 processes only one operation: $V_6 = \{v_{12}\}$.

Let the precedence relations between operations of the set V belonging to different jobs of the set J be given as follows: $v_1 \to v_{11}$; $v_6 \to v_3$; $v_8 \to v_3$; $v_7 \to v_{11}$; $v_8 \to v_{12}$; $v_9 \to v_4$; $v_9 \to v_{13}$; $v_{12} \to v_{15}$. These precedence relations determine the following set of arcs: $A \backslash A^* = \{(v_1, v_{11}), (v_6, v_3), (v_8, v_3), (v_7, v_{11}), (v_8, v_{12}), (v_9, v_4), (v_9, v_{13}), (v_{12}, v_{15})\}$ in the mixed graph $G = (V, A, E)$ such that implication (1) does not hold for each arc in the set $A \backslash A^*$.

Similarly to the mixed graph, representing input data of a shop-scheduling problem without operation preemptions [21; 28; 29; 32], input data for example 1 of the problem $G|t_i = 1|C_{\max}$ is given by the mixed graph $G = (V, A, E)$ depicted in fig. 1, where a set of all operations is represented by the vertex set $V = \bigcup_{i=1}^{|M|} V_i = \bigcup_{k=1}^{|J|} V^{(k)}$. The precedence constraints and capacity constraints are represented by a union of arc set A and edge set E.

Based on the above reduction of the general shop-scheduling problem $G|t_i = 1|C_{\max}$ to the colouring $c(G)$ of a suitable mixed graph $G = (V, A, E)$, one can derive the following correspondence of terms used in the optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ and terms used in the general shop-scheduling problem $G|t_i = 1|C_{\max}$:

$$\{\text{vertex } v_i \in V\} \Leftrightarrow \{\text{non-preemptive unit-time operation } v_i \in V\};$$

$$\{\text{vertices on path (on chain)} \left(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V^{(k)}|}}\right) \text{ in digraph } (V, A^*, \varnothing) \text{ (in graph } (V, \varnothing, E^*))\} \Leftrightarrow$$

$$\Leftrightarrow \{\text{set } V^{(k)} = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V^{(k)}|}}\right\} \text{ of linearly ordered operations of the job } J_k \in J\};$$

$$\{\text{precedence relations between operations belonging to different jobs}\} \Leftrightarrow \{\text{set of arcs } A \backslash A^* \text{ in digraph } (V, A \backslash A^*, \varnothing)\};$$

$$\{\text{clique } \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\} \text{ in graph } (V, \varnothing, E \backslash E^*)\} \Leftrightarrow \{\text{operations } V_i = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\} \text{ processed on machine } M_i \in M\};$$

{a colouring $c(G)$ of the mixed graph $G=(V, A, E)$} $\Leftrightarrow$ {a feasible schedule for the problem $G|t_i=1|C_{\max}$};
{an optimal mixed graph colouring $c(G)$} $\Leftrightarrow$ {an optimal schedule for the problem $G|t_i=1|C_{\max}$};
{the chromatic number $\chi(G)$} $\Leftrightarrow$ {the optimal value of makespan $C_{\max}$}.

The above correspondence of terms used in the optimal colouring $c(G)$ of the mixed graph $G=(V, A, E)$ and those used in the general shop-scheduling problem $G|t_i=1|C_{\max}$ implies the following claim.

Lemma 1. *Any general shop-scheduling problem $G|t_i=1|C_{\max}$ may be represented as an optimal mixed graph colouring $c(G)$ of a suitable mixed graph $G=(V, A, E)$.*

However, it is easy to see that an inverse claim to lemma 1 is not correct.

An optimal schedule for example 1 is determined by the following optimal colouring $c(G)$ of the mixed graph $G=(V, A, E)$: $c(v_1)=2$, $c(v_2)=4$, $c(v_3)=5$, $c(v_4)=1$, $c(v_5)=2$, $c(v_6)=3$, $c(v_7)=4$, $c(v_8)=5$, $c(v_9)=1$, $c(v_{10})=3$, $c(v_{11})=4$, $c(v_{12})=5$, $c(v_{13})=1$, $c(v_{14})=4$, $c(v_{15})=5$. This colouring $c(G)$ of the mixed graph $G=(V, A, E)$ is optimal, i. e., $\chi(G)=5$. Indeed, the optimality of the schedule determined by the mixed graph colouring $c(G)$ follows from the fact that there is a job $J_2 \in J$ consisting of five operations $V^{(2)}=\{v_4, v_5, v_6, v_7, v_8\}$, which implies the following non-strict inequality: $\chi(G)\geq 5$.

Job-shop scheduling with unit-time operations to minimise makespan. Article [2] with definition 1 of the mixed graph colouring $c(G)$ was published in Russian in 1976 along with other articles published before 1997. In 1997, another mixed graph colouring (called a strict mixed graph colouring $c_<(G)$) has been introduced in article [19] published in English.

Definition 2 [19]. An integer-valued function $c_<: V \to \{1, 2, \ldots, t\}$ is a strict colouring of the mixed graph $G=(V, A, E)$, if inequality $c_<(v_i)<c_<(v_j)$ holds for each arc $(v_i, v_j)\in A$ and $c(v_p)\neq c(v_q)$ for each edge $[v_p, v_q]\in E$. A strict mixed graph colouring $c_<(G)$ is optimal, if it uses a minimal possible number $\chi_<(G)$ of different colours $c_<(v_i)\in\{1, 2, \ldots, t\}$. A minimal number $\chi_<(G)$ is a strict chromatic number of the mixed graph $G=(V, A, E)$.

It is clear that one can use a colouring $c(G)$ (definition 1) instead of a strict colouring $c_<(G)$ (definition 2) for every specific mixed graph $G=(V, A, E)$ such that the following implication (2) holds for each arc $(v_i, v_j)\in A$:

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

Remark 1. A strict colouring $c_<(G)$ of the mixed graph $G=(V, A, E)$ is a special case of the colouring $c(G)$, if it is assumed that each inclusion $(v_i, v_j)\in A$ implies the inclusion $[v_i, v_j]\in E$ in the mixed graph $G=(V, A, E)$ to be coloured.

Due to remark 1, one can add edge $[v_i, v_j]$ to the mixed graph $G=(V, A, E)$ for each arc $(v_i, v_j)\in A$ such that implication (2) does not hold. Obviously, any strict colouring $c_<(G)$ of the mixed graph $G=(V, A, E)$ is a strict colouring $c_<(G^+)$ of the mixed graph $G^+=(V, A, E^+)$ constructed via adding all above edges $[v_i, v_j]$. Furthermore, strict mixed graph colourings $c_<(G)$ and $c_<(G^+)$ are the same as a mixed graph colouring $c(G^+)$.

The connection of the strict mixed graph colouring $c_<(G)$ and the job-shop scheduling problem $J|t_i=1|C_{\max}$ is studied in [4–8]. The job-shop scheduling problem $J|t_i=1|C_{\max}$ is a special case of the general shop-scheduling problem $G|t_i=1|C_{\max}$, if there are no precedence relations between operations belonging to different jobs (see [28–30]).

In article [4], it is shown that a mixed graph $G=(V, A, E)$ determining a job-shop scheduling problem $J|t_i=1|C_{\max}$ has the following mandatory properties.

Property 1. The partition $(V, \varnothing, E)=(V_1, \varnothing, E_1)\bigcup(V_2, \varnothing, E_2)\bigcup\ldots\bigcup(V_m, \varnothing, E_m)$ holds, where the subgraph $(V_k, \varnothing, E_k)$ of the mixed graph $G=(V, A, E)$ is a complete graph for each $k\in\{1, 2, \ldots, m\}$.

Property 2. The following partition $(V,\ A,\ \varnothing)=\left(V^{(1)},\ A^{(1)},\ \varnothing\right)\bigcup\left(V^{(2)},\ A^{(2)},\ \varnothing\right)\bigcup\ldots\bigcup\left(V^{(r)},\ A^{(r)},\ \varnothing\right)$ holds, where each directed subgraph $\left(V^{(k)},\ A^{(k)},\ \varnothing\right)$ of the mixed graph $G=(V,\ A,\ E)$ is a path $\left(v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{r_k}}\right)$ for $k\in\{1,\ 2,\ \ldots,\ r\}$.

Property 1 (property 2) means that the subgraph $(V,\ \varnothing,\ E)$ of the mixed graph $G=(V,\ A,\ E)$ is a union of disjoint complete graphs (the directed subgraph $(V,\ A,\ \varnothing)$ is a union of disjoint paths, respectively). In the job-shop scheduling problem $J|t_i=1|C_{\max}$, numbers m and r denote the cardinality of the machine set $M=\left\{M_1,\ M_2,\ \ldots,\ M_{|M|}\right\}$, $m=|M|$, and the cardinality of job set $J=\left\{J_1,\ J_2,\ \ldots,\ J_{|J|}\right\}$, $r=|J|$. Property 2 implies that if the inclusion $v_i\in V^{(k)}$ holds, operation v_i belongs to the job $J_k\in J$, and vice versa (see definition 2). A job $J_k\in J$ consisting of a set $V^{(k)}$ of linearly ordered operations is represented as path $\left(v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{r_k}}\right)$ in digraph $(V,\ A,\ \varnothing)$. Operations $V^{(k)}$ have to be processed in the order determined by path $\left(v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{r_k}}\right)$. Property 1 means that if the inclusion $v_i\in V_k$ holds, operation v_i has to be processed on machine $M_k\in M$. Due to definition 2 and property 1, each machine $M_k\in M$ can process at most one operation within any unit-time interval from the following set:

$$\{[0,\ 1],\ (1,\ 2],\ (2,\ 3],\ \ldots,\ (t-1,\ t]\}. \tag{3}$$

An optimal strict colouring $c_<:V\to\{1,\ 2,\ \ldots,\ \chi_<(G)\}$ of the mixed graph $G=(V,\ A,\ E)$ determines an assignment of operations V to a minimal number of the following intervals:

$$\{[0,\ 1],\ (1,\ 2],\ (2,\ 3],\ \ldots,\ (\chi_<(G)-1,\ \chi_<(G)]\}. \tag{4}$$

An assignment of operations V to the minimal number of unit-time intervals (4) is optimal since it determines a makespan optimal schedule for processing operations V, whose length is equal to the strict chromatic number $\chi_<(G)$ of the mixed graph $G=(V,\ A,\ E)$ determining an example of the unit-time minimum-length job-shop scheduling problem $J|t_i=1|C_{\max}$. Properties 1 and 2 define the usual assumptions used in scheduling theory [28–30] in terms of graph theory [26; 27]. The following lemma 2 is proved in article [4].

Lemma 2 [4]. *Any individual job-shop scheduling problem $J|t_i=1|C_{\max}$ is equivalent to an optimal strict colouring $c_<(G)$ of a suitable mixed graph $G=(V,\ A,\ E)$ possessing both properties 1 and 2, and vice versa.*

The proof of lemma 2 is based on the following correspondence of terms used in the strict mixed graph colouring $c_<(G)$ and those used in the job-shop problem $J|t_i=1|C_{\max}$:

{vertex $v_i\in V$} ⇔ {non-preemptive unit-time operation $v_i\in V$};

{vertices on path $\left(v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{1_{|V^{(k)}|}}}\right)$ in digraph $(V,\ A,\ \varnothing)$} ⇔ {set $V^{(k)}=\left\{v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{|V^{(k)}|}}\right\}$ of linearly ordered operations of the job $J_k\in J$};

{clique $\left\{v_{i_1},\ v_{i_2},\ \ldots,\ v_{i_{|V_i|}}\right\}$ in graph $(V,\ \varnothing,\ E)$} ⇔ {operations $V_i=\left\{v_{i_1},\ v_{i_2},\ \ldots,\ v_{i_{|V_i|}}\right\}$ processed on machine $M_i\in M$};

{a strict mixed graph colouring $c_<(G)$} ⇔ {a schedule for the problem $G|t_i=1|C_{\max}$};

{an optimal strict mixed graph colouring $c_<(G)$} ⇔ {an optimal schedule for the problem $G|t_i=1|C_{\max}$};

{the strict chromatic number $\chi_<(G)$} ⇔ {the optimal value of makespan $C_{\max}$}.

To illustrate lemma 2, we consider example 2 of the problem $J|t_i=1|C_{\max}$, which is the same as already considered example 1 of the problem $G|t_i=1|C_{\max}$ with only one exception that there is no precedence constraint between operations belonging to different jobs in the set J. In other words, it is assumed that $A\backslash A^*=\varnothing$. It is clear that a strict colouring $c_<(G)$ of the mixed graph $G=(V,\ A,\ E)$ depicted in fig. 2 determines a schedule existing for example 2. Obviously, the mixed graph $G=(V,\ A,\ E)$ depicted in fig. 2 possesses both properties 1 and 2. This mixed graph $G=(V,\ A,\ E)$ is a subgraph of the mixed graph depicted in fig. 1.

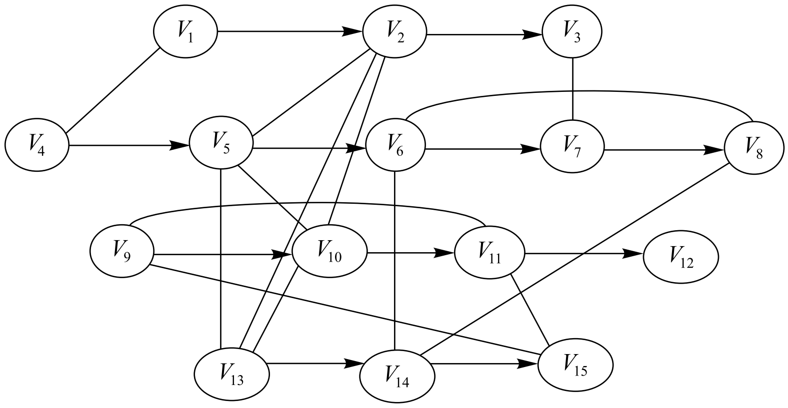

Fig. 2. Mixed graph $G = (V, A, E)$ determining example 2
of the problem $J|t_i = 1|C_{\max}$ with four jobs and six machines,
the optimal strict mixed graph colouring $c_<(G)$ being equivalent to example 2

An optimal schedule for example 2 is determined by the following strict mixed graph colouring $c_<(G)$: $c_<(v_1) = 2$, $c_<(v_2) = 4$, $c_<(v_4) = 1$, $c_<(v_6) = 3$, $c_<(v_7) = 4$, $c_<(v_8) = 5$, $c_<(v_9) = 1$, $c_<(v_{10}) = 3$, $c_<(v_{11}) = 4$, $c_<(v_{12}) = 5$, $c_<(v_{13}) = 1$, $c_<(v_{14}) = 4$, $c_<(v_{15}) = 5$. This strict colouring $c_<(G)$ is optimal, i. e., $\chi_<(G) = 5$, due to the existence of a job $J_2 \in J$ with five operations $V^{(2)} = \{v_4, v_5, v_6, v_7, v_8\}$ implying the following non-strict inequality: $\chi_<(G) \geq 5$.

It is important to highlight that there exists a general shop-scheduling problem $G|t_i = 1|C_{\max}$ which cannot be represented as the optimal strict colouring $c_<(G)$ of a mixed graph $G = (V, A, E)$. This shortage of a strict mixed graph colouring to represent a general shop-scheduling problem occurs since the strict inequality $c_<(v_i) < c_<(v_j)$ must hold for each arc $(v_i, v_j) \in A$ in the colouring $c_<(G)$, and therefore, a strict mixed graph colouring cannot define a precedence relation $v_i \to v_j$ on the operations v_i and v_j belonging to different jobs in the set J.

Remark 2. There are general shop-scheduling problems $G|t_i = 1|C_{\max}$, which cannot be represented as optimal strict colourings $c_<(G)$ of the suitable mixed graphs $G = (V, A, E)$.

In the following section, we introduce a new class of the scheduling problems that is more general than the classes of the problems $G|t_i = 1|C_{\max}$ and $J|t_i = 1|C_{\max}$ considered in this section. Based on the newly introduced class of the scheduling problems, we prove that an optimal colouring $c(G)$ of any colourable mixed graph $G = (V, A, E)$ is equivalent to an appropriate optimal unit-time minimum-length scheduling problem, and vice versa.

Unit-time scheduling partially ordered multi-processor tasks

Contrary to the scheduling problems studied in the previous section, where each operation has to be processed on a single machine, in the scheduling system with multi-processor tasks (MPT), a task may require either one processor (machine) or several processors during the complete period of processing the task [29; 33–36]. As usual, two tasks (operations) requiring at least one common processor (machine) cannot be processed simultaneously.

Chapter 10 of the book [29, p. 264–283] studies a general shop minimum-length scheduling problem $GMPT|t_i = 1|C_{\max}$ along with other scheduling problems $MPT|\beta|\gamma$ with multi-processor tasks [33–36]. The symbol G in the field α of the three-field notation $GMPT|t_i = 1|C_{\max}$ specifies a task system with arbitrary precedence constraints given on the set $V = \{v_1, v_2, \ldots, v_n\}$ of the multi-processor tasks. In the problem $GMPT|t_i = 1|C_{\max}$, it is needed to construct an optimal schedule for processing partially ordered multi-processor tasks $V = \{v_1, v_2, \ldots, v_n\}$ on the dedicated processors $M = \{M_1, M_2, \ldots, M_{|M|}\}$. The general shop-scheduling problem $G|t_i = 1|C_{\max}$ is a special case of the problem $GMPT|t_i = 1|C_{\max}$ since the processing of task $v_i \in V$ requires a single processor for the problem $G|t_i = 1|C_{\max}$. In the general shop-scheduling problem $GMPT|t_i = 1|C_{\max}$, a task $v_i \in V$ may be regarded as a job J_i including either one operation (task) $v_i \in V$ or more than one operation (several tasks from the set V). Let a simple job mean a job consisting only of one operation (task).

For any example of the problem $GMPT|t_i = 1|C_{\max}$, one can construct a mixed graph $G = (V, A, E)$ such that an optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ is equivalent to finding an optimal schedule for the problem $GMPT|t_i = 1|C_{\max}$. The construction of such a mixed graph $G = (V, A, E)$ is analogous to the construction of the mixed graph $G = (V, A, E)$ determining input data for the problem $G|t_i = 1|C_{\max}$ (see the previous section).

We next introduce a new class of the general shop-scheduling problems $GcMPT|t_i = 1|C_{\max}$, which includes the problem $GMPT|t_i = 1|C_{\max}$ as a special case studied in chapter 10 of the book [29]. More precisely, in the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$, it is required that a subset $V(k) = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\}$ of the tasks $V = \{v_1, v_2, \ldots, v_n\} \supseteq V(k)$ must be processed simultaneously in any feasible schedule. It is easy to see that the latter requirement may be represented by a circuit $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}, v_{k_1}\right)$ in the directed subgraph $(V, A_c, \varnothing)$ of the mixed graph $G = (V, A, E)$, which presents input data of the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$, where the set A of the arcs includes the following subset: $A_c' = \left\{\left(v_{k_1}, v_{k_2}\right), \left(v_{k_2}, v_{k_3}\right), \ldots, \left(v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}\right), \left(v_{k_{|V(k)|}}, v_{k_1}\right)\right\} \subseteq A.$

Let the input data for the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ include w subsets $V(1), V(2), \ldots, V(w)$ of the tasks such that every subset $V(k) = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\}$ of the tasks $V = \{v_1, v_2, \ldots, v_n\}$ must be processed simultaneously in any feasible schedule, where $k \in \{1, 2, \ldots, w\}$. Then, we determine the following subset of arcs:

$$A_c = \bigcup_{k=1}^{w} \left\{\left(v_{k_1}, v_{k_2}\right), \left(v_{k_2}, v_{k_3}\right), \ldots, \left(v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}\right), \left(v_{k_{|V(k)|}}, v_{k_1}\right)\right\}. \tag{5}$$

Similarly as in the previous section, one can establish the correspondence of terms used in the optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ with $A_c \subseteq A$ and those used in the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ see the table.

Obviously, every instance of the problem $GcMPT|t_i = 1|C_{\max}$ uniquely defines a mixed graph $G = (V, A, E)$ determining input data for this instance. Therefore, to describe an instance of the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$, it is sufficient to define a mixed graph $G = (V, A, E)$, which determines input data for this instance of the scheduling problem. In what follows, such an instance of the problem $GcMPT|t_i = 1|C_{\max}$ will be called the problem $GcMPT|t_i = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$.

The correspondence of terms used in the mixed graph colouring $c(G)$
and those used in the problem $GcMPT|t_i = 1|C_{\max}$ on mixed graph $G = (V, A, E)$

Terms of the mixed graph colouring $c(G)$	Terms of the problem $GcMPT|t_i = 1|C_{\max}$
Vertex $v_i \in V$	Unit-time task $v_i \in V$ (unit-time operation of the job)
Vertices on the path (on the chain, respectively) $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{1_{|V(k)|}}}\right)$ in the digraph $(V, A^*, \varnothing)$ (in the graph $(V, \varnothing, E^*)$)	Set $V^{(k)} = \left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\}$ of linearly ordered operations (tasks) of the job $J_k \in J$
Clique $\left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ in the graph $(V, \varnothing, E \backslash E^*)$	All tasks $V_i = \left\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\right\}$ processed on the same machine (processor) $M_i \in M$
Set of arcs $A \backslash A^*$ in the digraph $(V, A \backslash A^*, \varnothing)$	Precedence relations given between tasks (operations) belonging to different jobs of the set J

Ending table

Terms of the mixed graph colouring $c(G)$	Terms of the problem $GcMPT\|t_i=1\|C_{\max}$
Set of arcs $A\backslash A^*$ in the digraph $(V,\ A\backslash A^*,\ \varnothing)$	Precedence constraints given on the set of tasks V
Circuit $\left(v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{\|V(k)\|}},\ v_{k_1}\right)$ in the digraph $(V,\ A,\ \varnothing)$, where $A_c \subseteq A$	Tasks $V(k)=\left\{v_{k_1},\ v_{k_2},\ \ldots,\ v_{k_{\|V(k)\|}}\right\}\subseteq V$ that must be processed simultaneously
A mixed graph colouring $c(G)$ of the mixed graph $G=(V,\ A,\ E)$	A feasible schedule for the problem $GcMPT\|t_i=1\|C_{\max}$
An optimal mixed graph colouring $c(G)$ of the mixed graph $G=(V,\ A,\ E)$	An optimal schedule for the problem $GcMPT\|t_i=1\|C_{\max}$
The chromatic number $\chi(G)$	The optimal value of makespan $C_{\max}$

Due to the correspondence of terms used in the colouring $c(G)$ and those used in the equivalent problem $GcMPT|t_i=1|C_{\max}$ on the mixed graph $G=(V,\ A,\ E)$, one can derive lemma 3.

Lemma 3. *Every general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$ on the mixed graph $G=(V,\ A,\ E)$ is equivalent to an optimal mixed graph colouring $c(G)$.*

Contrary to the job-shop scheduling problem $J|t_i=1|C_{\max}$ having a feasible schedule for any input data, there are instances of the general shop-scheduling problem $G|t_i=1|C_{\max}$, which have no feasible schedules. To construct an instance of such an unsolvable individual general shop-scheduling problem $G|t_i=1|C_{\max}$, we add the precedence relation $v_5 \to v_9$ to the input data of example 1 depicted in fig. 1. We call this modified example as example 1^* and show that there is no feasible schedule for example 1^* due to the existence of the circuit $(v_4,\ v_5,\ v_9,\ v_4)$ in the digraph $(V,\ A,\ \varnothing)$ and the edge $[v_4,\ v_5]$ in the graph $(V,\ \varnothing,\ E^*)$. On the one hand, all tasks in the set $\{v_4,\ v_5,\ v_9\}$ must be processed simultaneously due to the circuit $(v_4,\ v_5,\ v_9,\ v_4)$. On the other hand, two tasks v_4 and v_5 cannot be processed simultaneously due to the edge $[v_4,\ v_5]\in E^*\subset E$. This contradiction implies that there is no feasible schedule for example 1^*. Since the general shop-scheduling problem $G|t_i=1|C_{\max}$ is a special case of the general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$, there are similar instances of the problem $GcMPT|t_i=1|C_{\max}$ such that no feasible schedules exist.

We prove the following criterion for the existence of a feasible schedule for the general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$ on the mixed graph $G=(V,\ A,\ E)$.

Theorem 2. *A feasible schedule for the general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$ on the mixed graph $G=(V,\ A,\ E)$ exists, if and only if the digraph $(V,\ A,\ \varnothing)$ has no circuit containing adjacent vertices in the graph $(V,\ \varnothing,\ E)$.*

P r o o f. Due to lemma 3, a general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$ on the mixed graph $G=(V,\ A,\ E)$ is equivalent to optimal colouring $c(G)$ of the mixed graph $G=(V,\ A,\ E)$. A mixed graph $G=(V,\ A,\ E)$ with $A\neq\varnothing$ and $E\neq\varnothing$ may be uncolourable, i. e., there is no colouring $c(G)$ for the mixed graph $G=(V,\ A,\ E)$. Furthermore, theorem 1 establishes a criterion for the existence of a colouring $c(G)$ for the mixed graph $G=(V,\ A,\ E)$ and this criterion directly proves theorem 2.

To illustrate lemma 3 and theorem 2, we consider two examples of the general shop-scheduling problem $GcMPT|t_i=1|C_{\max}$ with two non-simple jobs J_1 and J_2, eleven multi-processor tasks $V=\{v_1,\ v_2,\ \ldots,\ v_{11}\}$, seven machines $M=\{M_1,\ M_2,\ \ldots,\ M_7\}$, and three tasks $\{v_1,\ v_4,\ v_8\}$, which must be processed simultaneously in any feasible schedule. The mixed graph $G=(V,\ A,\ E)$ depicted in fig. 3 determines input data for example 3.

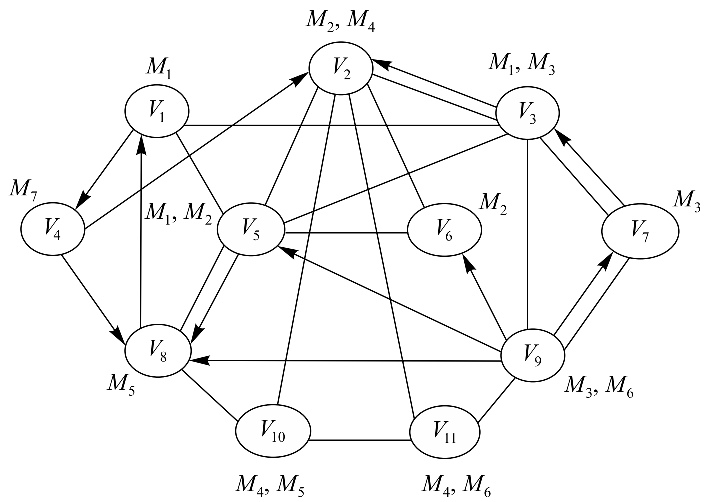

Fig. 3. Mixed graph $G = (V,\ A,\ E)$ determining
the problem $GcMPT|t_i = 1|C_{\max}$ with eleven tasks and seven machines,
the optimal mixed graph colouring $c(G)$ being equivalent to example 3

In example 3, machine M_1 has to process three tasks of the set $V_1 = \{v_1,\ v_3,\ v_5\}$. The forbiddance to process any pair of tasks from the set V_1 simultaneously is represented by the clique $\{v_1,\ v_3,\ v_5\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_2 has to process three tasks of the set $V_2 = \{v_2,\ v_5,\ v_6\}$. The forbiddance to process any pair of tasks from the set V_2 simultaneously is represented by the clique $\{v_2,\ v_5,\ v_6\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_3 has to process three tasks from the set $V_3 = \{v_3,\ v_7,\ v_9\}$. The forbiddance to process any pair of tasks from the set V_3 simultaneously is represented by the clique $\{v_3,\ v_7,\ v_9\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_4 has to process three tasks of the set $V_4 = \{v_2,\ v_{10},\ v_{14}\}$. The forbiddance to process any pair of tasks from the set V_4 simultaneously is represented by the clique $\{v_2,\ v_{10},\ v_{14}\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_5 has to process two tasks of the set $V_5 = \{v_8,\ v_{10}\}$. The forbiddance to process tasks from the set V_5 simultaneously is represented by the clique $\{v_8,\ v_{10}\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_6 has to process two tasks of the set $V_6 = \{v_9,\ v_{11}\}$. The forbiddance to process tasks from the set V_6 simultaneously is represented by the clique $\{v_9,\ v_{11}\}$ in the graph $(V,\ \varnothing,\ E \backslash E^*)$. Machine M_7 has to process one task: $V_7 = \{v_4\}$. All machines, which are used for processing the task $v_i \in V$, are presented near vertex v_i in fig. 3.

There are two jobs J_1 and J_2, which are not simple. Job $J_1 \in J$ consists of the set $V^{(1)} = \{v_9,\ v_7,\ v_3,\ v_2\}$ of linearly ordered tasks (operations). Thus, job $J_1 \in J$ is represented by a union of the path $(v_9,\ v_7,\ v_3,\ v_2)$ in the digraph $(V,\ A^*,\ \varnothing)$ and the chain $(v_9,\ v_7,\ v_3,\ v_2)$ in the graph $(V,\ \varnothing,\ E^*)$. Job $J_2 \in J$ consists of the set $V^{(2)} = \{v_5,\ v_8\}$ of linearly ordered tasks (operations). Thus, job $J_1 \in J$ is represented by a union of the path $(v_5,\ v_8)$ in the digraph $(V,\ A^*,\ \varnothing)$ and the chain $(v_5,\ v_8)$ in the graph $(V,\ \varnothing,\ E^*)$. Other precedence relations between tasks and operations belonging to different non-simple jobs are determined as follows: $v_1 \to v_4$; $v_4 \to v_2$; $v_4 \to v_8$; $v_8 \to v_1$; $v_9 \to v_5$; $v_9 \to v_6$; $v_9 \to v_8$. All the precedence constraints determine the subset A of the arcs in the digraph $(V,\ A \backslash A^*,\ \varnothing)$, where $A \backslash A^* = \{(v_1,\ v_4),\ (v_4,\ v_2),\ (v_5,\ v_3),\ (v_4,\ v_8),\ (v_8,\ v_1),\ (v_9,\ v_5),\ (v_9,\ v_4),\ (v_9,\ v_6),\ (v_9,\ v_8)\}$.

In example 3, every job J_i from the set $J \backslash \{J_1,\ J_2\}$ is simple, i. e., job J_i consists of a single task that is identified with job J_i. In example 3, the set of tasks $V^{(1)} = \{v_1,\ v_4,\ v_8\} \subset V$ must be processed simultaneously in any feasible schedule. This requirement is determined by the circuit $(v_1,\ v_4,\ v_8,\ v_1)$ in the digraph $(V,\ A_c,\ \varnothing)$. Based on the correspondence of the terms (see the table), we construct the mixed graph $G = (V,\ A,\ E)$ depicted in fig. 3, which determines the input data for example 3 of the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ with eleven multi-processor tasks, seven machines, two non-simple jobs, and three tasks, which must be processed simultaneously in any feasible schedule.

Due to theorem 2, there exists a feasible schedule for example 3 of the problem $GcMPT|t_i = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$. Due to lemma 3, an optimal schedule for example 3 may be determined by the following optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ depicted in fig. 3: $c(v_1) = 2$, $c(v_2) = 4$, $c(v_3) = 3$, $c(v_4) = 2$, $c(v_5) = 1$, $c(v_6) = 2$, $c(v_7) = 2$, $c(v_8) = 2$, $c(v_9) = 1$, $c(v_{10}) = 1$, $c(v_{11}) = 2$. This mixed graph colouring $c(G)$ is optimal due to lemma 3 since $\chi(G) = 4$. Indeed, the optimality of the schedule determined by the mixed graph colouring $c(G)$ follows from the fact that there exists a job $J_1 \in J$ consisting of four unit-time operations $V^{(1)} = \{v_9, v_7, v_3, v_2\}$ that imply the non-strict inequality $\chi(G) \geq 4$.

Due to theorem 2, there are examples of the general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ such that no feasible schedules exist for them. We construct an example of such an unsolvable general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ as follows. We replace the precedence relation $v_9 \to v_8$ by the opposite relation $v_8 \to v_9$ in the input data of example 3 depicted in fig. 3. There is no feasible schedule for such a modified example 3 (we call it example 3^*), where the precedence relation $v_9 \to v_8$ is replaced by the relation $v_8 \to v_9$. On the one hand, all tasks from the set $\{v_5, v_8, v_9\}$ must be processed simultaneously due to the circuit (v_5, v_8, v_9, v_5). On the other hand, two tasks v_5 and v_8 cannot be processed simultaneously due to the edge $[v_5, v_8] \in E^* \subset E$. The contradiction obtained along with theorem 2 implies that there is no feasible schedule for example 3^* of the problem $GcMPT|t_i = 1|C_{\max}$.

We next prove the following lemma, which is the inverse of lemma 3.

Lemma 4. *For any colourable mixed graph* $G = (V, A, E)$*, there exists a general shop-scheduling problem* $GcMPT|t_i = 1|C_{\max}$ *on the mixed graph* $G = (V, A, E)$*, which is equivalent to finding an optimal colouring* $c(G)$.

P r o o f. We detect a set Ω of all circuits existing in the directed subgraph $(V, A, \varnothing)$ of the mixed graph $G = (V, A, E)$ and consider two possible cases: either $\Omega = \varnothing$ or $\Omega \neq \varnothing$.

Case 1. Let the set Ω be empty; $\Omega = \varnothing$. Then, one can construct the desired problem $GcMPT|t_i = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$ using the following algorithm.

Algorithm

Input: a mixed graph $G = (V, A, E)$ such that no circuit exists in the digraph $(V, A, \varnothing)$.

Output: a general shop-scheduling problem $GcMPT|t_i = 1|C_{\max}$ on the mixed graph $G = (V, A, E)$, which is equivalent to finding an optimal colouring $c(G)$.

Step 1: partition the graph $(V, \varnothing, E)$ into (maximal) components as follows:

$$(V, \varnothing, E) = (V_1, \varnothing, E_1) \bigcup \ldots \bigcup (V_m, \varnothing, E_m) \bigcup (V_{m+1}, \varnothing, \varnothing) \bigcup \ldots \bigcup (V_{m+r}, \varnothing, \varnothing),$$

where the subgraph $(V_k, \varnothing, E_k)$ is a (maximal) component of the graph $(V, \varnothing, E)$ for each $k \in \{1, \ldots, m\}$ such that $|V_k| \geq 2$. The subgraph $(V_j, \varnothing, \varnothing)$ determines an isolated vertex for each index $j \in \{m+1, \ldots, m+r\}$. Denote this isolated vertex as follows: $\{v_{j_1}\} := V_j$. Set $M = \varnothing$, $k = 1$, $i = 0$, $l_0 = 0$.

Step 2: IF $k = m + 1$ **THEN GOTO** step 5 **ELSE** find all maximal (relative to inclusion) complete vertex-induced subgraphs $(V_k^1, \varnothing, E_k^1), \ldots, (V_k^{l_k}, \varnothing, E_k^{l_k})$ of the connected graph $(V_k, \varnothing, E_k)$. Set $r = 1$, $i := i + l_{k-1} + 1$.

Step 3: FOR index i, supplement machine M_i with the already constructed machine set, i. e., $M := M \cup \{M_i\}$. Establish that all tasks in the clique V_k^r of the connected graph $(V_k, \varnothing, E_k)$ must be processed on machine M_i, i. e., $V_k^r = V_i = \{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\}$, where all tasks $\{v_{i_1}, v_{i_2}, \ldots, v_{i_{|V_i|}}\}$ must be processed on machine M_i in any feasible schedule. Set $i := i + 1$.

Step 4: IF $i = \sum_{h=0}^{k} l_h$ **THEN** set $k := k + 1$ **GOTO** step 2 **ELSE** set $r := r + 1$ **GOTO** step 3.

Step 5: FOR each index $j \in \{m+1, \ldots, m+r\}$, supplement machine M_{i+j-m} with the already constructed machine set M. Establish that task v_{j_1} with $V_j = \{v_{j_1}\}$, which is isolated in the graph $(V, \varnothing, E)$, must be processed on machine M_{i+j-m}. Establish that machine M_{i+j} must process only task v_{j_1}. Set $M := M \cup \{M_{i+1}, \ldots, M_{i+r}\}$.

Step 6: FOR each arc $\left(v_p, v_q\right)$ existing in the directed subgraph $(V, A, \varnothing)$ of the mixed graph $G = (V, A, E)$, introduce the precedence relation $v_p \to v_q$, which means that processing the task v_p must be completed before starting the task v_p in any feasible schedule.

Step 7: a general shop-scheduling problem $GcMPT\left|t_i = 1\right|C_{\max}$ is constructed on the mixed graph $G = (V, A, E)$, where the precedence relations on the task set V are determined at step 6 and the machine set M is determined at step 3 and step 5 **STOP**.

Case 2. Let the set Ω be not empty, Ω = ∅. Since the mixed graph $G = (V, A, E)$ is colourable, every circuit $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}, v_{k_1}\right)$ in set Ω has no adjacent vertices in the subgraph $(V, \varnothing, E)$ of the mixed graph G (theorem 1). Therefore, all tasks $\left\{v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}\right\} =: V(k)$ must be processed simultaneously in any feasible schedule for the desired general shop-scheduling problem $GcMPT\left|t_i = 1\right|C_{\max}$ on the mixed graph $G = (V, A, E)$, where the circuit $\left(v_{k_1}, v_{k_2}, \ldots, v_{k_{|V(k)|}}, v_{k_1}\right)$ exists in the directed subgraph $(V, A, \varnothing)$.

Let $\Omega = \bigcup_{k=1}^{w} V(k) = \bigcup_{k=1}^{w} \left\{\left(v_{k_1}, v_{k_2}, v_{k_3}, \ldots, v_{k_{|V(k)|-1}}, v_{k_{|V(k)|}}, v_{k_1}\right)\right\}$. Then, we delete all arcs A_c defined in (5) from the mixed graph $G = (V, A, E)$ and apply the above algorithm to the obtained circuit-free mixed graph $G^0 = (V, A \backslash A_c, E)$. As a result, the problem $GcMPT\left|t_i = 1\right|C_{\max}$ is constructed on the mixed graph $G^0 = (V, A \backslash A_c, E)$, which is equivalent to finding an optimal colouring $c\left(G^0\right)$. Consequently, the problem $GcMPT\left|t_i = 1\right|C_{\max}$ on the mixed graph $G = (V, A, E)$ is equivalent to finding an optimal colouring $c(G)$. Lemma 4 is thus proved.

We apply the above constructive proof of lemma 2 to the mixed graph $G = (V, A, E)$ depicted in fig. 3 and obtain example 4 of the problem $GcMPT\left|t_i = 1\right|C_{\max}$ on the mixed graph $G = (V, A, E)$ depicted in fig. 4. Note that examples 3 and 4 are different, e. g., all jobs are simple in example 4, while there are two non-simple jobs in example 3.

In general, it is easy to show that the proof of lemma 4 implies that for any colourable mixed graph $G = (V, A, E)$, one can construct a general shop-scheduling problem $GcMPT\left|t_i = 1\right|C_{\max}$ on the mixed graph $G = (V, A, E)$, which is equivalent to finding an optimal colouring $c(G)$ and all jobs in the set J are simple.

Note that it is required to use non-simple jobs for some objective functions $\gamma = f\left(C_1, C_2, \ldots, C_{|J|}\right)$, since only completion times C_i of the jobs $J_i \in J$ are their arguments. Hence, the completion times of some tasks may be ignored in the values of such objective functions. In particular, the total completion time $\gamma = \Sigma C_i = \sum_{i=1}^{|J|} C_i$ is such an objective function.

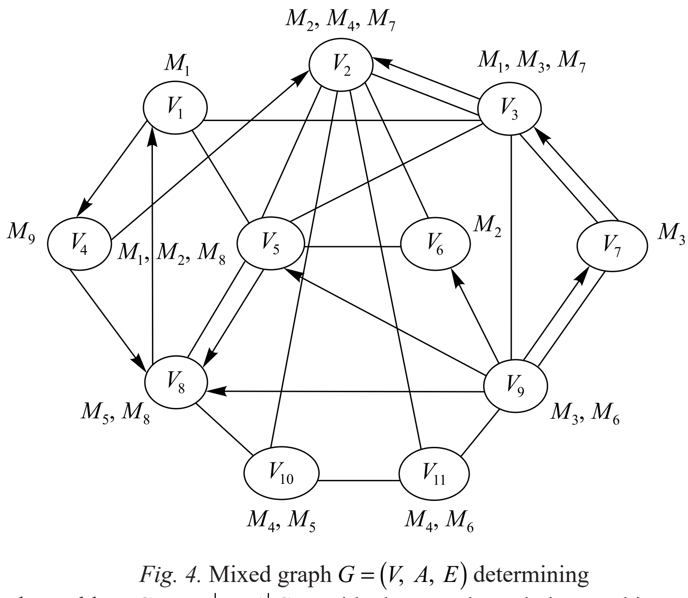

Fig. 4. Mixed graph $G = (V, A, E)$ determining the problem $GcMPT\left|t_i = 1\right|C_{\max}$ with eleven tasks and nine machines, the optimal mixed graph colouring $c(G)$ being equivalent to example 4

Let $T = \{\tau_1, \tau_2, \ldots, \tau_{|T|}\}$ denote a set of all tasks (operations) in the scheduling problem $GcMPT | t_i = 1 | C_{\max}$, and let $C(\tau_i)$ denote a completion time of the task $\tau_i \in T$. Due to the equalities $\max\{C_1, C_2, \ldots, C_{|J|}\} = C_{\max} = \max\{C(\tau_1), C(\tau_2), \ldots, C(\tau_{|T|})\}$, the completion time of any task cannot be ignored in the value of the objective function $\gamma = C_{\max}$. Therefore, only simple jobs may be used in the shop-scheduling problem with minimising makespan $C_{\max}$, where any non-simply job may be represented by the precedence relations on the task set T and the completion time $C(\tau_i)$ of each task $\tau_i \in T$ cannot be ignored in the value of $C_{\max}$.

Obviously, the following theorem combines lemmas 3 and 4.

Theorem 3. *Every general shop-scheduling problem $GcMPT | t_i = 1 | C_{\max}$ on the mixed graph $G = (V, A, E)$ is equivalent to finding an optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$. Further, for any colourable mixed graph $G = (V, A, E)$, there exists an individual general shop-scheduling problem $GcMPT | t_i = 1 | C_{\max}$ on the mixed graph $G = (V, A, E)$, which is equivalent to finding an optimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$.*

To restrict a set of feasible schedules for a shop-scheduling problem $\alpha|\beta|\gamma$, which must be tested in order to minimise a value of the regular objective function γ [15], a finite set of semi-active schedules may be considered, since there exists an optimal semi-active schedule for a shop-scheduling problem $\alpha|\beta|\gamma$ with any regular objective function γ [28].

Definition 3 [28; 29]. A schedule is called semi-active, if no task (operation) can be processed earlier without violating a given constraint or changing the task (operation) processing order in the obtained schedule.

We remark that any colouring $c(G)$ of the mixed graph $G = (V, A, E)$ uniquely determines a strict order on the colours $c(v_j)$ of all vertices in the set V. Due to this remark, one can define a minimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$ as follows.

Definition 4. A colouring $c(G)$ of the mixed graph $G = (V, A, E)$ is called minimal, if no colour $c(v_i)$ can be decreased without changing the order of colours $c(v_j)$ of the vertices in the set $V \setminus \{v_i\}$ in the obtained colouring $c'(G)$ of the mixed graph $G = (V, A, E)$.

Obviously, each semi-active schedule existing for the general shop-scheduling problem $GcMPT | t_i = 1 | C_{\max}$ on the mixed graph $G = (V, A, E)$ uniquely determines a minimal colouring $c(G)$ of the mixed graph $G = (V, A, E)$, and vice versa. Hence, we obtain theorem 4.

Theorem 4. *There exists a one-to-one correspondence between all minimal colourings $c(G)$ of the mixed graph $G = (V, A, E)$ and all semi-active schedules existing for the general shop-scheduling problem $GcMPT | t_i = 1 | C_{\max}$ on the mixed graph $G = (V, A, E)$.*

We used both graph terminology and scheduling one for the above problems. However, it is possible to describe most presented results either using only graph terminology or using only scheduling terminology.

Conclusion

We introduced a new class of general shop-scheduling problems $GcMPT | t_i = 1 | C_{\max}$ for finding optimal schedules for partially ordered multi-processor tasks with unit processing times. Contrary to a classical shop-scheduling problem, several machines are required to process a task in the problem $GcMPT | t_i = 1 | C_{\max}$. It is also required that a subset of tasks must be processed simultaneously in any feasible schedule. We proved theorem 3 showing that an optimal colouring $c(G)$ of any mixed graph $G = (V, A, E)$ is equivalent to the general shop-scheduling problem $GcMPT | t_i = 1 | C_{\max}$, and vice versa. Hence, many terms used in scheduling theory (such as schedule, job, machine, processor, operation, task, processing time and makespan) may be considered as usual terms used in mixed graph colourings $c(G)$.

Due to theorems 3 and 4, most results that have been proven so far [4; 5; 7; 8; 14; 28; 29; 33–36] and will be proven in future for the scheduling problem $GcMPT | p_i = 1 | C_{\max}$ and for its special cases have analogous results for optimal colourings $c(G)$ of the appropriate mixed graphs $G = (V, A, E)$. Conversely, most results that have been proven so far [2; 4; 5; 7; 8; 19; 25–27] and will be proven in future for optimal mixed graph colourings $c(G)$ have analogous results for scheduling problems $GcMPT | t_i = 1 | C_{\max}$.

References

1. Wan L, Mei J, Du J. Two-agent scheduling of unit processing time jobs to minimize total weighted completion time and total weighted number of tardy jobs. *European Journal of Operational Research.* 2021;290(1):26–35. DOI: 10.1016/j.ejor.2020.07.064.
2. Sotskov YN, Tanaev VS. [A chromatic polynomial of a mixed graph]. *Izvestiya Akademii nauk BSSR. Seriya fiziko-matematicheskikh nauk.* 1976;6:20–23. Russian.
3. Karp RM. Reducibility among combinatorial problems. In: Miller RE, Thatcher JW, Bohlinger JD, editors. *Complexity of computer computations*. Boston: Springer; 1972. p. 85–103. DOI: 10.1007/978-1-4684-2001-2_9.
4. Sotskov YN, Dolgui A, Werner F. Mixed graph coloring for unit-time job-shop scheduling. *International Journal of Mathematical Algorithms.* 2001;2:289–323.
5. Sotskov YN, Tanaev VS, Werner F. Scheduling problems and mixed graph colorings. *Optimization.* 2002;51(3):597–624. DOI: 10.1080/0233193021000004994.
6. Al-Anzi FS, Sotskov YN, Allahverdi A, Andreev GV. Using mixed graph coloring to minimize total completion time in job shop scheduling. *Applied Mathematics and Computation.* 2006;182(2):1137–1148. DOI: 10.1016/j.amc.2006.04.063.
7. Kouider A, Ait Haddadène H, Ourari S, Oulamara A. Mixed graph colouring for unit-time scheduling. *International Journal of Production Research*. 2017;55(6):1720–1729. DOI: 10.1080/00207543.2016.1224950.
8. Kouider A, Ait Haddadène 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. *Computer & Operations Research*. 2019;4967:1001–1008.
9. Lenstra JK, Rinnooy Kan AHG. Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics.* 1979;4:121–140. DOI: 10.1016/S0167-5060(08)70821-5.
10. Sotskov YN. Complexity of optimal scheduling problems with three jobs. *Cybernetics.* 1990;26(5):686–692. DOI: 10.1007/BF01068549.
11. Sotskov YN. The complexity of shop-scheduling problems with two or three jobs. *European Journal of Operational Research*. 1991;53(3):326–336. DOI: 10.1016/0377-2217(91)90066-5.
12. Sotskov YN, Shakhlevich NV. NP-hardness of shop-scheduling problems with three jobs. *Discrete Applied Mathematics*. 1995; 59(3):237–266. DOI: 10.1016/0166-218X(95)80004-N.
13. Kravchenko SA, Sotskov YN. Optimal makespan schedule for three jobs on two machines. *Mathematical Methods of Operations Research*. 1996;43(2):233–238. DOI: 10.1007/BF01680374.
14. Gonzalez T. Unit execution time shop problems. *Mathematics of Operations Research*. 1982;7(1):57–66. DOI: 10.1287/moor.7.1.57.
15. Brucker P, Kravchenko SA, Sotskov YN. On the complexity of two machine job-shop scheduling with regular objective functions. *OR Spektrum*. 1997;19:5–10. DOI: 10.1007/BF01539799.
16. Shakhlevich NV, Sotskov YN. Scheduling two jobs with fixed and nonfixed routes. *Computing*. 1994;52(1):17–30. DOI: 10.1007/BF02243393.
17. Shakhlevich NV, Sotskov YN, Werner F. Shop-scheduling problems with fixed and non-fixed machine orders of the jobs. *Annals of Operations Research*. 1999;92:281–304. DOI: 10.1023/A:1018943016617.
18. Damaschke P. Parameterized mixed graph coloring. *Journal of Combinatorial Optimization.* 2019;38:326–374. DOI: 10.1007/s10878-019-00388-z.
19. Hansen P, Kuplinsky J, de Werra D. Mixed graph colorings. *Mathematical Methods of Operations Research*. 1997;45:145–160. DOI: 10.1007/BF01194253.
20. Kruger K, Sotskov YN, Werner F. Heuristics for generalized shop scheduling problems based on decomposition. *International Journal of Production Research*. 1998;36(11):3013–3033. DOI: 10.1080/002075498192265.
21. Sotskov YN. Software for production scheduling based on the mixed (multi)graph approach. *Computing & Control Engineering Journal.* 1996;7(5):240–246. DOI: 10.1049/CCE:19960509.
22. Sotskov YN. Mixed multigraph approach to scheduling jobs on machines of different types. *Optimization*. 1997;42(3):245–280. DOI: 10.1080/02331939708844361.
23. de Werra D. Restricted coloring models for timetabling. *Discrete Mathematics*. 1997;165–166:161–170. DOI: 10.1016/S0012-365X(96)00208-7.
24. de Werra D. On a multiconstrained model for chromatic scheduling. *Discrete Applied Mathematics.* 1999;94(1–3):171–180. DOI: 10.1016/S0166-218X(99)00019-0.
25. Sotskov YN. Mixed graph colorings: a historical review. *Mathematics.* 2020;8(3):385. DOI: 10.3390/math8030385.
26. Harary F. *Graph theory*. Massachusetts: Addison-Wesley; 1969. 274 p.
27. Thulasiraman K, Swamy MNS. *Graphs: theory and algorithms*. Canada: John Wiley & Sons, Inc.; 1992. 480 p. DOI: 10.1002/9781118033104.
28. Tanaev VS, Sotskov YN, Strusevich VA. *Scheduling theory. Multi-stage systems.* Dordrecht: Kluwer Academic Publishers; 1994. 406 p. DOI: 10.1007/978-94-011-1192-8.
29. Brucker P. *Scheduling algorithms.* Berlin: Springer; 1995. 326 p. DOI: 10.1007/978-3-662-03088-2.
30. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics.* 1979;5:287–326. DOI: 10.1016/S0167-5060(08)70356-X.
31. Sotskov YN, Tanaev VS. Scheduling theory and practice: Minsk group results. *Intelligent Systems Engineering.* 1994;3(1):1–8. DOI: 10.1049/ISE.1994.0001.
32. Sussmann B. Scheduling problems with interval disjunctions. *Mathematical Methods of Operations Research.* 1972;16:165–178.
33. Brucker P, Krämer A. Shop scheduling problems with multiprocessor tasks on dedicated processors. *Annals of Operations Research.* 1995;57(1):13–27. DOI: 10.1007/BF02099688.
34. Brucker P, Krämer A. Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems. *European Journal of Operational Research.* 1996;90(2):214–226. DOI: 10.1016/0377-2217(95)00350-9.
35. Hoogeveen JA, van de Velde SL, Veltman B. Complexity of scheduling multiprocessor tasks with prespecified processor allocations. *Discrete Applied Mathematics.* 1994;55(3):259–272. DOI: 10.1016/0166-218X(94)90012-4.
36. Hoogeveen JA, Lenstra JK, Veltman B. Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *European Journal of Operational Research.* 1996;89(1):172–175. DOI: 10.1016/S0377-2217(96)90070-3.

Received 21.03.2021 / revised 27.06.2021 / accepted 27.06.2021.