Abstract. A theoretical analysis of a new optimal scheduling approach with an application to a flexible flow shop with continuous flows and uniform alternative machines is considered. The peculiarity of the considered problem is the simultaneous consideration of both continuous flows in the operations and discrete assignments. The scheduling approach is based on a dynamic non-stationary interpretation of the execution of the operations and a temporal decomposition of the scheduling problem. The algorithmic realization is based on a modified form of the continuous maximum principle blended with mathematical optimization. The novelty of this study as compared with previous works on this method consists of a detailed theoretical analysis of the temporal decomposition and computational complexity with an application to flow-shop scheduling with continuous flows and discrete assignments. The optimality conditions as well as the structural properties of the model and the algorithm are investigated. Advantages and limitations of the proposed approach are discussed. The results of this study can be used as an extension to the existing mathematical programming models subject to the following issues: dynamics and non-stationarity of the operation execution, non-stationarity of the machine structures and capacity parameters, and representation of continuous flows. In addition, the formulation of the scheduling model in terms of control makes it possible to attract additional tools from mathematics of functional spaces such as stability, robustness, controllability, adaptability, etc. to the schedule analysis and real-time adjustment.

Keywords: scheduling, flexible flow shop, continuous flows, alternative machines, optimal program
control, mathematical programming.

Abbreviations:
IP – integer programming
LP – linear programming
MP – mathematical programming
MSA – method of successive approximations
OPC – optimal program control
QI – quality index

1 Introduction

We study a multi-stage scheduling problem, where the operations have continuous flows but the transition between the stages is discrete. In scheduling theory, considerable achievements can be stated regarding assignment and sequencing problems in the context of multi-stage systems (Blazewicz et al. 2001, Lauff & Werner, 2004). At the same time, most papers on multi-stage systems deal with problems, where the computational complexity represents the most critical challenge and mathematical programming (MP) models are typically solved with heuristic algorithms (Chiou, Chen, Liu, & Wu, 2012). Another challenge in the multi-stage systems with alternative machines for each stage is the different processing speed at the machines which may influence the processing (task) times (Kyparisis & Koulamas, 2006).

The peculiarity of the problem under consideration is the simultaneous consideration of both continuous flows in the operations and discrete assignments, i.e., the problem is discrete-continuous. In these settings, an interesting constellation of discrete and continuous elements can be observed (Shah 2005). On the one hand, an assignment problem is discrete by nature and requires the introduction of binary variables, i.e., discrete optimization techniques can be correctly used here. At the same time, a non-stationary execution for continuous flows can be correctly described in terms of continuous optimization (Shah and Ierapetritou 2012, Subramanian et al. 2013).

The basic theory for studying multi-stage, multi-period dynamic systems with continuous variables and performance indicators accumulated over time is optimal control theory in general, and optimal program control (OPC) in particular (Athaus & Falb, 1966; Lee & Markus, 1967; Sethi & Thompson, 2000). At the same time, a direct application of OPC to a pure combinatorial problem is questionable. For this reason, the basic technical idea of the approach considered in this paper is to apply the methods of discrete optimization to combinatorial tasks within certain time intervals and to use the OPC for (1) the continuous flow representation within the operations and (2) interlinking the solutions over the planning horizon. Since the assignment and flow control problems are interrelated (i.e., the task time is needed for the solution of an assignment problem, but it is calculated in the flow control model; in turn for the task time calculation in a flow control model, the assignment decisions are needed), they will be considered simultaneously in this study. An additional peculiarity of such simultaneous consideration is that both the machine structures and the flow parameters may be uncertain and change in dynamics and are, therefore, non-stationary.
In order to exemplify the application, this study considers the multi-stage flexible flow shop scheduling problem with uniform parallel machines (Kyparisis & Koulamas, 2006) where job splitting is allowed (Tahar et al. 2006). In addition, we consider continuous flows, a multi-objective formulation, and non-stationary processing speeds at the machines. The novelty of this study as compared with previous works on this method (Shah and Ierapetritou 2012, Subramanian et al. 2013) consists of a detailed theoretical analysis of the time-based decomposition and the computational complexity with an application to flow-shop scheduling with continuous flows and discrete assignments.

Consider the problem statement (the full list of notations is given in Appendix 1). At each $l$-stage, some uniform alternative machines $M^{(j)}$ from the set $M = \{M^{(j)}, j \in N, N = (1, \ldots, n)\}$ exist. At each $l$-stage, some uniform alternative machines $M^{(j)}$ exist. Consider the operations $D^{(i)}$ from the set $D = \{D^{(i)}, \mu \in \bar{S}, \bar{S} = (1, \ldots, \mu)\}$, each of which belongs to a job $\bar{B}^{(i)}$ from the set $\bar{B} = \{\bar{B}^{(i)}, i \in N, N = (1, \ldots, n)\}$. The independent jobs consist of a chain of operations. All jobs are assumed to be available for processing at time 0. Each machine $M^{(j)}$ is capable of producing all the operations at the $l$-stage, but it can handle only one job at a time. Note that for a simplification, the stage index $l$ is omitted in the further progress of this paper, and it is assumed to be considered in the machine indexes $j$ subject to the non-stationary machine availability at each stage (i.e., “availability windows”), which is expressed in the preset matrix time function $\varepsilon_{\mu}(t)$. The structure considered is presented in Figure 1.

At each of the stages, each machine $M^{(j)}$ has a speed (i.e., an effective processing intensity) $c_{ij}$ that is subject to the total machine capacity $\overline{R}_j$. The impact of the speed $c_{ij}(t)$ is that the machine $M^{(j)}$ can process $a_{ij}$ units subject to the planned processing volume $a_{ij}$ and $c_{ij}(t)$. An operation $D^{(i)}$ may start only after the previous operation $D^{(i-1)}$ has been completed. All jobs have to be completed by time $T_f$. For a simplification of the model and the algorithm presentation in this paper, we allow pre-emption and do not consider setups (i.e., in Figure 1, the operations may be partially executed on different machines at each of the stages as known from the job splitting literature (e.g., Tahar et al. 2006)).

The problem consists of scheduling the operations taking into account flow dynamics control subject to three objectives: $J_1$ – minimization of total lateness (subject to $T_f$), $J_2$ –
maximization of the volume of the fully completed jobs (subject to \( a_{ij} \) and \( a_{ij} \); i.e., in the ideal case \( a_{ij} = a_{ij} \) for all jobs subject to \( e_{ij}(t) \) and \( e_{ij}(t) \); a strong requirement on the full completion of all jobs by the time \( T_f \) may also be included), and \( J_3 \) – equal utilization of the machines (subject to \( \tilde{R}_j \)).

The considered problem contains many practical peculiarities which are rarely considered in the literature in an integrated manner and therefore can be interesting for many managerial decisions on scheduling. In particular, the managers are always interested in non-deterministic approaches to scheduling where scheduling is interconnected to the control function (Maccarthy, & Liu, 1993). In the existing papers, many non-deterministic and control elements are included into consideration. For example, the machines are not always available and can break down. Not all jobs are available at the commencement of processing. Pre-emption is allowed. The processing times and technological constraints are non-deterministic and fluctuate in time.

The remainder of this paper is organized as follows. Section 2 is devoted to a literature review. Section 3 presents the methodology. In Section 4, the model is described and a scheduling problem in terms of OPC is stated. Section 5 analyses the computational procedure. In Section 6, the optimality, existence, and complexity are analyzed. The paper concludes by discussing the main findings and some future research works.

2 Literature review

Beginning with the work of Johnson (1954), flow shop scheduling models (i.e., models where a set of jobs must be processed on a number of machines sequentially disposed and the operations for different jobs have to be processed in the same order) have been extensively considered in the literature, including, for instance, Gupta, Neppalli, and Werner (2001); Kubzin, Potts, and Strusevich (2009); Dugardin, Yalaoui, and Amodeo (2010); Ribas, Leisten, and Framiñan (2010). Most of the authors minimize the makespan or earliness/tardiness costs (Huq et al. 2004, Li et al. 2009). Since the problems of this class, with a few exceptions, have been proved to be NP-hard (Gonzales & Sahni, 1978), heuristic solutions are predominantly applied in practice (Jungwattanakit, Reodecha, Chaovalitwongse, & Werner, 2008, Jungwattanakit, Reodecha, Chaovalitwongse, & Werner, 2009, Laha and Sarin 2009, Martin 2009, Werner 2013).

Scheduling with alternative parallel machines has also been a large research avenue over the past few decades. In such systems, the goal is to optimize both the selection of machines for each part and the loading sequences of the parts to the machines to improve the productivity (Hankins, Wvsk, & Fox, 1984) or to minimize the makespan (Weglarz, 1976). Blazewicz, Dror, and Weglarz’s (1991) review showed that these problems are NP-hard. Józefowska et al. (2002) presented a heuristic approach to allocating a continuous resource in discrete–continuous scheduling problems to minimize the makespan. In some cases, e.g., for identical processors with unit processing times, such a scheduling problem can be transformed to a transportation problem and solved with integer programming (IP) methods (Graham, Lawler, Lenstra, & Rinnooy Kan, 1979). Kyparis and Koulamas (2006) considered a multi-stage
flexible flow shop scheduling problem with uniform parallel machines at each stage and makespan minimization. This study proposed a heuristic schedule computation for this strongly NP-hard problem. Tazar et al. (2006) considered the problem of scheduling a set of independent jobs with sequence-dependent setup times and job splitting on a set of identical parallel machines such that the maximum completion time (makespan) is minimized. For this NP-hard problem, this study developed a heuristic algorithm using linear programming (LP).

Consideration has also been given to continuous flows which are typical for the processing industry, e.g., petrochemistry, energy supply, oil and gas industries (Shah 2005; Mujawar et al. 2012; Subramanian, Rawlings, Maravelias, Flores-Cerrillo, & Megan 2013). These networks are typically centralized and of multi-stage nature, i.e., the jobs consist of a logically arranged sequence of operations. A practical challenge is that at each stage alternative machines may perform the operations. This creates flexibility in the process plan (Yu, Liu, Wang, & Fan, 2011) and requires both a machine assignment and sequencing the tasks (Patel, Shah, & Ashe, 2011). The optimization objectives in practice are multiple ones and may be related to the maximization of the processed jobs, completing the jobs on time (e.g., minimization of maximal lateness), and an equal machine utilization over time.

Let us turn to the OPC applications. The studies by Holt, Modigliani, Muth, and Simon (1960), Hwang, Fan, and Erikson (1967), Zimin and Ivanilov (1971) and Moiseev (1974) were among the first to apply the OPC and the maximum principle to multi-level and multi-period master production scheduling that determined the production as an optimal control with a corresponding trajectory of the state variables (i.e., the inventory). This stream was continued by Kimemia and Gershwin (1983), who applied a hierarchical method in designing a solution procedure to the overall model, and by Khmelnitsky, Kogan, and Maimom (1997) for planning continuous-time flows in flexible manufacturing systems.

The study (Sarimveis, Patrinos, Tarantilis, & Kiranoudis, 2008) showed a wide range of advantages regarding the application of OPC to production and logistics. They include, first of all, a non-stationary process view and accuracy of continuous time. In addition, a wide range of analysis tools from control theory regarding stability, controllability, adaptability, etc. may be used if a schedule is described in terms of control. Recent studies (e.g., Subramanian, Rawlings, Maravelias, Flores-Cerrillo, & Megan 2013) discussed the possibilities to translate the MP scheduling models into a state-space form and the design of rescheduling algorithms with the desired closed-loop properties.

However, although the OPC was widely applied to flexible manufacturing system scheduling, it cannot be directly applied to the flow or job shop scheduling level as a computational procedure. The continuous time models are not applicable in their direct form to discrete assignment problems due to the continuous values of the control variables from 0 to 1. In addition, such problems as numerical instability, non-existence of gradients, and non-convexity of state space should be mentioned. The calculation of the OPC with direct methods of the continuous maximum principle has also not been proved to be efficient. It can be concluded that the application of OPC to scheduling is not a trivial problem for two reasons. First, a conceptual problem consists of the continuous values of the control variables. Second, a computational problem with a direct method of the maximum principle exists. These
shortcomings set limitations on the application of OPC to purely combinatorial problems.

At the same time, in some problems (e.g., discrete assignments and continuous flows, non-stationarity of machine and capacity parameters, large-scale assignment matrix), the elements of OPC can extend the existing MP scheduling techniques. For this reason, the idea of this study is not to use OPC for solving the combinatorial problem but rather to enhance the existing MP algorithms regarding the non-stationarity, flow control, and continuous material flows. In these settings, two hypotheses can be formulated.

- If the control variables could be presented as binary variables (and not as a continuous control), it might become possible to incorporate them into the assignment problem.
- In using a modified maximum principle, the OPC can be used for a dynamic decomposition and aggregation in flexible flow shop scheduling models.

The basic technical idea of our approach, which extends the previous application of the maximum principle to production and logistics, is to apply methods of discrete optimization to combinatorial tasks within certain time intervals and to use the OPC with all its advantages (i.e., accuracy of continuous time, integration of planning and control, and the operation execution parameters as time functions) for (1) the flow control within the operations and (2) interlinking the partial (decomposed) solutions into an optimal schedule.

In the proposed approach, OPC is mainly used for the dynamic decomposition but not for the calculations. The computational procedure is transferred to the MP methods. The solution at each time point is calculated with MP. OPC is used for modeling the execution of the operations and interlinking the MP solutions over the planning horizon. Hence, the solution procedure becomes independent of the continuous optimization algorithms and can be of discrete nature, e.g., an integer assignment model.

3 Methodical approach

The basic conceptual idea of this approach is the fact that the operation execution and machine availability are dynamically distributed in time over the planning horizon. As such, not all operations and machines are involved in the decision making at the same time. Therefore, it becomes quite natural to transit from large-size allocation matrices with a high number of binary variables to a scheduling problem that is dynamically decomposed.

Following an approach to decompose the solution space and to use exact methods over its restricted sub-spaces, we propose to use the OPC theory for the dynamic decomposition of the scheduling problem. The computational procedure will be based on a modified maximum principle in the continuous form blended with MP. One of the basic problems in applying any decomposition method is how then to evaluate the overall system performance. By using the maximum principle, this problem is solved a priori. Since the maximum principle guarantees that the optimal solutions of the instantaneous problems give an optimal solution to the overall problem (Pontryagin, Boltyanskiy, Gamkrelidze, & Mishchenko, 1964; Boltyanskiy, 1973; Sethi & Thompson, 2000), this principle is a convenient approach to a problem decomposition into a number of sub-problems. For these sub-problems, optimal solutions can be found, e.g., with the help of MP. Then these solutions are linked into an OPC.
The original dynamic interpretation of the assignment of the operations to a non-stationary set of machines can be exemplified in the following way. Consider four machines and six jobs $\bar{B}^{(i)}$ each of which is composed of 3–6 operations $D_{\mu}^{(i)}$. At each time instant, only one operation can be processed on one machine (see Figures 2–5). Note that in this example, for a simplification, we consider only a one-stage system with four alternative uniform machines. In the model, a multi-stage flow shop will be considered (see Fig. 1).
Figures 2–5 Dynamic representation of the scheduling model
In Figures 2–5, the execution dynamics of six jobs is presented for four time instants \( t = t_1, t_2, t_3, t_f \). Note that in this example of a continuous problem statement, \( t = t_1, t_2, t_3, t_f \) corresponds to \( l = 1, 2, 3, 4 \) in the discrete case as presented in Figure 1. Different colors describe current execution states. The operations marked in black have already been completed. The operations marked in gray may be executed subject to the machine availability and precedence relations. The operations marked in white cannot be executed yet because of the precedence relations. For example, at \( t = t_2 \), the operation \( D_2^{(4)} \) cannot be assigned since the operation \( D_1^{(4)} \) is still being processed with the use of the machine \( M_1^{(4)}(u_{1,4}^{(1)}(t_2) = 1) \).

From Figures 2–5, it can be observed that at each time instant, the assignment decisions consider only the gray colored operations subject to some available (“competing”) machines. The assignment of a machine \( M^{(j)} \) to the execution of the operation \( D_\mu^{(i)} \) can be described by the piecewise continuous function \( u_{\mu j}^{(i)}(t) \) that becomes equal to 1 in the case of an assignment. In the following course of this paper (see Section 4), these functions will play the role of OPC within the proposed dynamic model of the execution of the operations.

In the upper part of Figures 2–5, the machines \( M^{(j)} \) are listed which are available at the current time instant. At \( t = t_1, t_2, t_f \), the four machines \( M^{(1)}, M^{(2)}, M^{(3)}, M^{(4)} \) are available. At \( t = t_3 \), only the three machines \( M^{(1)}, M^{(2)}, M^{(4)} \) are available. This non-stationary set of machine availability will be represented in the model (see Section 5) with the help of the preset matrix time function \( e_{ij}(t) \).

The main advantage of the proposed approach to the problem class is the consideration of real event logic in the flow shop. The flow shop scheduling problem becomes naturally decomposed according to the logic of time. In the existing optimal scheduling methods, time aspects are represented as quite limited. For example, within MP, for the scheduling horizon \( t = [T_0, T_f] \), all possible assignments are represented within one system of algebraic equalities or inequalities. In the proposed dynamic interpretation of the execution of the operations, the large-scale multi-dimensional combinatorial matrix is decomposed. It can be observed from Figures 2–5 that the current dimensionality of the considered scheduling problem for \( t = t_1, t_2, t_3, t_f \) is determined by the dimensionality of the gray colored area. The operations in the black and white areas are not considered at the given time points and, therefore, will not influence the mathematical model for the assignment. If so, the OPC and MP models can be integrated subject to their advantages. The calculation procedure is transferred to MP methods and is therefore independent from the OPC. The solution at each point of time is calculated with MP. OPC is used for modeling the execution of operations and interlinking the MP solutions over the planning horizon.
4 Model

The formal statement of the scheduling problem will be produced, as it has been noted above, via a dynamic interpretation of the execution processes of the operations. In the remainder of this section, we will consider the partial dynamic models in more detail.

4.1 Mathematical model for the control processes of the operations (model $M_o$)

Let us consider the mathematical model for processing the operation $D^{(i)}_\mu$ in the job $\overline{B}^{(i)}$. The following notations can be introduced:

- $x^{(o)}_{ij\mu}$ is a variable characterizing the state of the operation $D^{(i)}_\mu$, where $^{(o)}$ indicates the relationship of the state variable $x$ to the operation states.
- $\varepsilon_j(t)$ is the given preset matrix time function of the time-spatial constraints;
- $u^{(o)}_{ijg}(t)$ is the decision control action at the moment $t$.

$t$ is the current time instant; $t \in T = (T_0, T_f]$ is the planning horizon,

$T_0 (T_f)$ is the start and end time instant of the planning horizon.

The dynamics of the operation $D^{(i)}_\mu$ can be expressed as follows:

$$\frac{dx^{(o)}_{ij\mu}}{dt} = \dot{x}^{(o)}_{ij\mu} = \sum_{j=1}^{n} \varepsilon_j(t) u^{(o)}_{ijg}(t).$$  \hspace{1cm} (1)

Eq. (1) represents the operation execution dynamics in which the non-stationarity of the execution of the operations is reflected. We have $\varepsilon_j(t) = 1$, if machine $M^{(j)}$ is available, and $\varepsilon_j(t) = 0$, otherwise (e.g., a constraint on the production shift from 7am to 4pm). $u^{(o)}_{ijg}(t)$ is a decision variable. We have $u^{(o)}_{ijg}(t) = 1$ at the time point $t$, if the operation $D^{(i)}_\mu$ is assigned to the machine $M^{(j)}$, and $u^{(o)}_{ijg}(t) = 0$ otherwise.

The continuous time allows to represent the execution of the operations at each time point, and therefore, to obtain additional information about the execution of the operations. An example of a control profile for the execution of the operations is given in Figure 6.

![Graph showing the execution dynamics of the operations](image)
In Figure 6, an example of a control profile for the execution of one operation on one machine is presented. The usage of continuous time allows a non-stationary analysis of the schedule execution (e.g., the machine availability between 8:00 and 10:00 am, or between 9:00 and 10:00 am may have a different impact on the schedule performance). The state variable \( x(t) \) accumulates the executed (processed) volume of the considered operation. Assuming that the planned execution volume is 6 units, it can be observed from Figure 6 that the given operation can be fully executed (i.e., \( a_{ij}^{(0)} = x_{ij}^{(0)} \)) on the given machine and completed on time (by \( T_f \)) with a processing time of 13 time units. In this case, the control variable \( u_{ij}^{(0)}(t) \) (Eq. 1) will switch to 1, which means that the assignment is possible. Such principles are equivalent to those used in the further models.

The control actions are constrained as follows:

\[
\sum_{i=1}^{n} \sum_{\mu=1}^{s_i} u_{ij}^{(0)}(t) \leq 1, \quad j = 1, ..., n; \quad \sum_{j=1}^{n} u_{ij}^{(0)}(t) \leq 1, \quad i = 1, ..., n; \quad \mu = 1, ..., s_i, \quad (2)
\]

\[
\sum_{j=1}^{n} u_{ij}^{(0)} \left[ \sum_{a \in I_{\mu i}} (a_{i\alpha}^{(0)} - x_{i\alpha}^{(0)}) + \prod_{\beta \in I_{\mu j}} (a_{i\beta}^{(0)} - x_{i\beta}^{(0)}) \right] = 0, \quad (3)
\]

\[
u_{ij}^{(0)}(t) \in \{0, 1\}, \quad (4)
\]

where conditions (3) - (5) hold for \( i = 1, ..., n; \quad \mu = 1, ..., s_i; \quad j = 1, ..., n; \quad \Gamma_{i\mu}, \Gamma_{ij} \) are the sets of operations which immediately precede the operation \( D_{\mu i} \), \( \sum_{j=1}^{n} u_{ij}^{(0)} \sum_{a \in I_{\mu i}} (a_{i\alpha}^{(0)} - x_{i\alpha}^{(0)}) = 0 \) is an “and” constraint, which denotes the condition of the total processing of all the predecessor operations, \( \sum_{j=1}^{n} u_{ij}^{(0)} \prod_{\beta \in I_{\mu j}} (a_{i\beta}^{(0)} - x_{i\beta}^{(0)}) = 0 \) is an “or” constraint, which denotes the condition of the processing of at least one of the predecessor operations.

Constraints (2) define the 1x1 assignment problem. Constraints (3) bring the natural time logic into the model and determine the precedence relations by blocking the operation \( D_{\mu i} \) until the previous operations \( D_a^{(0)}, D_{\beta}^{(0)} \) have been completed.

In order to assess the results of the execution of the operations, we define the following start and end conditions:

\[
h_0^{(0)}(x^{(0)}(T_0)) \leq 0; \quad h_1^{(0)}(x^{(0)}(T_f)) \leq 0, \quad (5)
\]

where \( h_0^{(0)}, h_1^{(0)} \) are known differentiable functions that determine the start and end conditions of the vector

\[
x^{(0)} = (x_1^{(0)}, ..., x_{\mu_i})^T. \quad (6)
\]
The initial and end conditions (7) and (8) specify the values of the variables at the beginning and end of the planning period, namely:

at the moment \( t = T_0 \) : \( x_{i\mu}^{(o)}(T_0) = 0 \); \hfill (7)

at the moment \( t = T_f \) : \( x_{i\mu}^{(o)}(T_f) = a_{i\mu}^{(o)} \). \hfill (8)

The OPC \( \mathbf{u}(t) \) and the state trajectory \( \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \) should be determined so that the constraints (7) and (8) are met; in other words, the desired values of the performance indicators should be achieved as an analogy to goal programming.

Constraint (7) reflects that, at the beginning, the volume of the executed operations is equal to zero (in the case that a certain volume of the orders is to be transferred from the previous planning period to the beginning of the current planning period, this should be reflected in (8)). Condition (8) reflects the desired end state, i.e., the completion of the operations by the time \( T_f \).

According to the problem statement, let us introduce the following performance indicators (objectives):

\[
J_1^{(o)} = \frac{1}{2} \sum_{i=1}^{n} \sum_{\mu=1}^{s_i} (a_{i\mu}^{(o)} - x_{i\mu}^{(o)}(T_f))^2 ,
\]

\[
J_2^{(o)} = \sum_{i=1}^{n} \sum_{\mu=1}^{s_i} \int_{T_0}^{T_f} \alpha_{i\mu}^{(o)}(\tau) u_{i\mu}^{(o)}(\tau) d\tau .
\]

The performance indicator \( J_1^{(o)} \) (function (9)) characterizes the accuracy of the accomplishment of the end conditions, i.e., the volume of the completed operations by the time \( T_f \). This can also express the extent of losses caused by a non-fulfillment of the end conditions. The objective function (10) minimizes total maximum lateness using penalties. The function \( \alpha_{i\mu}^{(o)}(\tau) \) is assumed to be known for each operation.

4.2 Mathematical model for the machine control (model \( \mathbf{M}_k \))

Let us introduce equation (11) to assess the total machine availability time:

\[
\frac{d\dot{x}_j^{(k)}}{dt} = \ddot{x}_j^{(k)} = \sum_{i=1}^{n} \sum_{\mu=1}^{s_i} u_{i\mu j}^{(o)}(t) ,
\]

where \( (k) \) indicates the relationship of the state variable \( x \) to the machines. Equation (11) represents the machine utilization. The variable \( \dot{x}_j^{(k)} \) characterizes the total employment time of machine \( \mathbf{M}_j^{(k)} \).

The end conditions are identical to Eqs. (5)–(8).

According to the problem statement, let us introduce the following performance indicator:

\[
J_1^{(k)} = \frac{1}{2} \sum_{j=1}^{m} (T - x_j^{(k)}(T_f))^2 .
\]
The indicator $J^{(k)}_i$ helps to estimate the uniformity of the machine use at the end point $t = T_f$ of the planning period. For example, in the supply chain scheduling context, this may be a requirement for a supply chain collaboration.
4.3 Mathematical model for the flow control (model $M_f$)

The interrelations and mutual impacts of the assignment and flow control still remain an open research question. In the proposed approach, these decisions are considered simultaneously. Recall that the task times may differ regarding different speeds $c_{ijg}(t)$ and machine availabilities $\varepsilon_g(t)$. For instance, the assignment of an operation from Figure 6 to another machine could result in a different execution control profile and task time. For this reason, the assignments from the model $M_o$ (made on the basis of the volumes $a_{\mu i}$) are now subject to further optimization regarding the flow dynamics control.

An assignment of an operation to a machine and the starting execution of the operations causes dynamic flows of the processed products. Let us introduce a model for the flow dynamics control (13):

$$\frac{dx_{ijg}^{(f)}}{dt} = x_{ijg}^{(f)} = u_{ijg}^{(f)},$$  \hspace{1cm} (13)

where $^{(f)}$ indicates the relationship of the state variable $x$ to the flows. The economic sense of Eq. (13) consists in the dynamic representation of the material flows resulting from the execution of the operations on the machine $M^{(j)}$. The meaning of Eq. (13) is very close to a system dynamics model to balance the flows in a system. However, the proposed approach also considers the strictly defined logic of the execution of the operations (model $M_o$). Moreover, the models of operations and flow control are interlinked linearly by Eq. (14) and the conjunctive system (27)–(29).

In contrast to model $M_o$, the control variable $u_{ijg}^{(f)}(t)$ is not a binary variable, but is equal to the processed flow volume $x_{ijg}^{(f)}(t)$ at each time point $t$. The model $M_f$ uses the assignment results from the model $M_o$ in the form of the control variables $u_{ijg}^{(o)}(t)$ and extends them by the actual processing speed of the machines subject to the following constraints:

$$0 \leq u_{ijg}^{(f)}(t) \leq c_{ijg}^{(f)} \cdot u_{ijg}^{(o)},$$  \hspace{1cm} (14)

$$\sum_{i=1}^{k} \sum_{j=1}^{s} u_{ijg}^{(f)}(t) \leq \tilde{R}_j^{(f)} \cdot \xi^{(f)}(t),$$  \hspace{1cm} (15)

Inequalities (14) use the assignment decisions ($u_{ijg}^{(o)}(t)$) from the model $M_o$ and the processing speed $c_{ijg}(t)$ of the machines $M^{(j)}$ for the optimization problems (13)–(18).

Constraints (15) reflect that the processing speed is constrained by $\tilde{R}_j^{(f)}$ taking into account the lower and upper bounds of some perturbation impacts $0 \leq \xi^{(f)}(t) \leq 1$ which may decrease the capacity availability.

Remark: Eqs. (15) and (16) set up the attainable set (AS) of the OPC in a dynamic system, i.e., all possible states of the schedule execution subject to different variations of the
parameters (e.g., the capacity availability). The introduction of Eq. (15) allows to analyse feasible schedule executions under conditions of non-stationary perturbations. If so, an AS can be used to analyze the schedule robustness i.e., an ability to continue the schedule execution subject to the specified objectives in the presence of perturbations.

The end conditions in $M_f$ are identical to (5)–(8).

According to the problem statement, let us introduce the following objectives:

$$J_{1}^{(f)} = \frac{1}{2} \sum_{i=1}^{\pi} \sum_{\mu=1}^{s_i} \sum_{j=1}^{n} (x_{i\mu j}^{(f)} - x_{i\mu j}^{(f)}(T_f))^2, \quad (16)$$

$$J_{2}^{(f)} = \frac{1}{2} \sum_{i=1}^{\pi} \sum_{\mu=1}^{s_i} \sum_{j=1}^{n} \int_{t_0}^{T_f} \beta_{i\mu j}^{(f)}(\tau) u_{i\mu j}^{(f)}(\tau) d\tau. \quad (17)$$

The economic meaning of the objectives (16)–(17) is identical to the objectives (9)–(10). The function $\beta_{i\mu j}^{(f)}(\tau)$ is assumed to be known for each operation.

Remark 2. Note that the constraints (2)–(4) and (14)–(15) are identical to those in the MP models. However, at each $t$-point of time, the number of variables in the calculation procedure is determined by the operations, which are currently in the “active zone” of scheduling, i.e., the operations marked in gray in Figures 2–5. For the problem sizes subject to the “active zone”, known methods for the solution of the MP models (e.g., the Hungarian method for $M_o$ or linear programming (LP) for $M_f$) can be applied to the problem (1)–(17).

4.4 Formulation of the scheduling problem

The models described above can be presented in an integrated form (model $M$). As mentioned above, the model should provide the decision makers with alternatives to handle. The performance indicators may be weighted in different proportions depending on the planning goals and operational strategies. The preference relations (minmax, maxmin, etc.) form the Pareto space and allow the calculation of a general relative quality index (QI) (18) within the corresponding schedule $u(t)$.

$$\min J(x(t), u(t), \xi(t), t) = \left\| \begin{bmatrix} J_{1}^{(o)}, J_{2}^{(o)}, J_{1}^{(k)}, J_{1}^{(f)}, J_{2}^{(f)} \end{bmatrix} \right\|, \quad (18)$$

where $J_{1}^{(o)}, J_{2}^{(o)}, J_{1}^{(k)}, J_{1}^{(f)}, J_{2}^{(f)}$ are the values of the performance indicators.

For a simplification, it is assumed that the transition from the vector form $J$ to a scalar form $J_G$ has been performed on the basis of the weight coefficients $\lambda_{i}^{(o)}, \lambda_{2}^{(o)}, \lambda_{3}^{(k)}, \lambda_{4}^{(f)}, \lambda_{5}^{(f)}$.

Now the scheduling problem can be formulated as the following problem of dynamic system control. The task is to find a feasible control $u(t)$, $[T_0, T_f]$ which ensures that the dynamic control model meets the constraint functions and guides the dynamic system (i.e., the schedule) $\dot{x} = f(t, x, u)$ from the initial state to the specified final state subject to given end conditions and the uncertainty area under the disturbances $\xi(t)$. If there are several feasible
controls (schedules), then the best one (optimal) should be selected in order to maximize (minimize) the components of $J_G$. We refer to this problem as $PS$.

5 Computational procedure and analysis of the algorithm

The computational procedure for the developed model is based on the integration of the main and conjunctive equation systems subject to the maximization of the following Hamiltonian (19)–(21):

$$H(x^*(t), u^*(t), \psi(t)^*) = \max_{\tilde{u} \in Q(x)} \sum_{z=1}^2 H_z(x(t), u(t), \psi(t)). \quad (19)$$

$$H_1 = \sum_{l=1}^{n_1} \sum_{i=1}^{n} \sum_{j=1}^{n} [\psi_{ij}^{(0)} \cdot \xi_i + \psi_{ij}^{(k)} + w_{j}^{(o)} \alpha_{ijg}^{(o)}] u_{ijg}^{(o)}, \quad (20)$$

$$H_2 = \sum_{l=1}^{n_1} \sum_{i=1}^{n} \sum_{j=1}^{n} [\psi_{ijg}^{(0)} + w_{j}^{(f)} \beta_{ijg}^{(f)}] u_{ijg}^{(f)}, \quad (21)$$

where $\psi(t)$ is the conjunctive vector.

The maximization of the Hamiltonian $H_1$ for model (1) in combination with the constraints (2)–(4) solves the assignment problem. The maximization of the Hamiltonian $H_2$ for model (13) in combination with the constraints (14)–(15) solves the LP problem. At each time instant, only those jobs and constraints from the “active scheduling zone” (i.e., the area marked in gray in Figures 2–5) are considered in the models $M_o$ and $M_f$ which meet the requirements (2)–(4), (5), (14), and (15). By a dynamic switching of the constraints (3) from inequalities to equalities, the size of the scheduling problem at each time point is reduced. The Hamiltonians (20) and (21) can be maximized when the constraints (3) satisfy the corresponding variables $u_{ijg}^{(o)}$ and $u_{ijg}^{(f)}$. In this case, only a part of the constraints (3) and (14) is considered for the current assignment problem since, when the control in (3) is switched to zero, then it becomes active in the right-hand part of the equations (14). Therefore, the reduction of the problem dimensionality at each time instant in the calculation process is ensured due to the recurrent operation description.

Theorem 1 Let $\Lambda$ be a relaxed problem for the problem $PS$. Then:

a) If the problem $\Lambda$ does not have a feasible solution, then this is true for the problem $PS$ as well.

b) If the OPC of the problem $\Lambda$ is feasible, then it is the OPC for the problem $PS$ as well.

Proof

a) If the problem $\Lambda$ does not have a feasible solution, then the control $u(t)$ that transfers the dynamic systems (1)–(4) and (11), (13)–(15) $\dot{x} = f(x,u,t)$ from a given initial state to a given final state does not exist.
b) Let \( u^*(t), \forall t \in (T_0, T_f] \), be an OPC in \( \Lambda \) and \( x(t) \) be a solution to model \( M \) subject to \( u(t) = u^*(t) \). Then \( u^*(t) \) meets the requirements of the local cut method and maximizes the Hamiltonian for the problem \( PS \). Hence, the vectors \( u^*(t) \) and \( x^*(t) \) return the minimum to the performance indicators (9)–(10), (12), (16)–(17). The proof is complete.

**Corollary 1** The PS problem can be transferred to a boundary problem.

**Proof.** As the dynamics of the state and conjunctive variables \( x(t) \) and \( \psi(t) \) is described by the differential equations, it becomes possible to calculate \( x(t) \) and \( \psi(t) \) at any time instant subject to the given initial conditions \( (x_0, \psi_0) \). The Hamiltonian (19) turns into a function of only one variable \( u(t) \) that can be calculated at any \( t \) subject to \( u(t) \in Q_\alpha \). Therefore, the PS problem can be reduced to a two-point boundary problem with the help of the local cut method (Boltianskiy, 1973). The proof is complete.

A methodical challenge in applying the maximum principle is to find the coefficients of the conjunctive system which change in dynamics. One of the contributions of this paper is that these coefficients can be found analytically from Eqs. (23) and (24). The coefficients of the conjunctive system play the role of the dynamical Lagrange multipliers as compared with MP dual formulations.

The conjugate system can be written as follows (Boltianskiy, 1973; Moiseev, 1974):

\[
\dot{\psi}_i = -\frac{\partial H}{\partial x_i} + \sum_{a=1}^{\tilde{n}} \delta_a(t) \frac{\partial q_a^{(1)}(x(t), u(t))}{\partial x_i} + \sum_{\beta=1}^{\tilde{r}} \rho_\beta(t) \frac{\partial q_\beta^{(2)}(x(t), u(t))}{\partial x_i} .
\]  

(22)

The coefficients \( \delta_a(t), \rho_\beta(t) \) can be determined by means of the following expressions (23)–(24):

\[
\rho_\beta(t) q_\beta^{(2)}(x(t), u(t)) = 0, \quad \bar{\beta} \in \{1, \ldots, \tilde{r}_2\} , \quad \text{(23)}
\]

\[
\text{grad}_\alpha H(x(t), u(t), \psi(t)) = \sum_{a=1}^{\tilde{n}} \delta_a(t) \text{grad}_\alpha q_a^{(1)}(x(t), u(t)) + \sum_{\beta=1}^{\tilde{r}} \rho_\beta(t) \text{grad}_\beta q_\beta^{(2)}(x(t), u(t)) .
\]  

(24)

In the formulas (22)–(24), \( x_i \) are the elements of the general state vector \( x(t) \) and \( \psi(t) \) are the elements of the conjugate vector \( \psi(t) \). In accordance with the maximum principle, the following conjugate system can be written (25)–(27):

\[
\frac{d\psi_i^{(k)}}{dt} = \psi_j^{(k)} = 0 ,
\]  

(26)

\[
\frac{d\psi_j^{(f)}}{dt} = \psi_{ij}^{(f)} = 0 .
\]  

(27)
The transversality conditions can be formulated in the following way (28)–(30):

\[ \psi_{ij}^{(o)}(T_f) = \lambda_{i}(a_{ij}^{(o)} - x_{ij}^{(o)}(T_f)) \]  
(28)

\[ \psi_{j}^{(k)}(T_f) = \lambda_{j}(T - x_{j}^{(k)}(T_f)) \]  
(29)

\[ \psi_{ij}^{(f)}(T_f) = \lambda_{ij}(a_{ij}^{(f)} - x_{ij}^{(f)}(T_f)) \]  
(30)

Let us consider the algorithmic realization of the above-described modified maximum principle. After transforming it into a boundary problem, a relaxed problem can be solved to receive an OPC, for the computation of which the main and conjunctive systems are integrated, i.e., the OPC vector \( \mathbf{u}^*(t) \) and the state trajectory \( \mathbf{x}^*(t) \) are obtained. The OPC vector at time \( t = T_0 \) and for the given value of \( \psi(t) \) should return the maximum to (9)–(10), (12), and (16)–(17), which have been transformed to a general performance index and expressed in scalar form \( J_G \).

The basic peculiarity of the boundary problem considered is that the initial conditions for the conjunctive variables \( \psi(t_0) \) are not given. At the same time, an OPC should be calculated subject to the end conditions (5)–(8). To obtain the conjunctive system vector, we use the Krylov–Chernousko method of successive approximations (MSA) for an OPC problem with a free right end which is based on the joint use of a modified successive approximation method (Krylov & Chernousko, 1972). We propose to use a heuristic schedule \( \mathbf{u}(t) \) to obtain the initial conditions for \( \psi(t_0) \). Then, the algorithm DYN can be stated as follows:

**Step 1** An initial solution \( \mathbf{u}(t), t \in [T_0, T_f] \) (a feasible control, in other words, a feasible schedule) is selected and \( r = 0 \).

**Step 2** As a result of the dynamic model run, \( \mathbf{x}^{(r)}(t) \) is received. Besides, if \( t = T_f \) then the record value \( J_G = J_G^{(r)} \) can be calculated. Then, the transversality conditions (30)–(32) are evaluated.

**Step 3** The conjugate system (27)–(29) is integrated subject to \( \mathbf{u}(t) = \mathbf{u}(t) \) and over the interval from \( t = T_f \) to \( t = T_0 \). For the time \( t = T_0 \), the first approximation \( \psi^{(r)}(T_0) \) is obtained as a result. Here, the iteration number \( r = 0 \) is completed.

**Step 4** From the time point \( t = T_0 \) onwards, the control \( \mathbf{u}^{(r+1)}(t) \) is determined \( (r = 0, 1, 2, \ldots) \) denotes the number of the iteration). In parallel with the maximization of the Hamiltonian, the main system of equations and the conjugate one are integrated. The maximization involves the solution of several MP problems at each time point.

The assignments (i.e., the control variables \( u_{ij}^{(o)} \)) from the model \( \mathbf{M}_o \) are used in the flow control \( \mathbf{M}_f \) (13)–(17) by means of the constraints (14). At the same time, the model \( \mathbf{M}_f \) influences the model \( \mathbf{M}_o \) through the transversality conditions (28)–(30), the conjunctive system (25)–(27), and the Hamiltonian function (19). In addition, the possible machine
structure dynamics and flow control dynamics through perturbation impacts is taken into account in (11) and (15).

In each iteration, the main part of the functional $J_G$ is negative and has a maximal absolute value among the main parts of increments computed for all possible variations of the control variables. In contrast to gradient methods and classical formulas of calculus of variations, it is sufficient to use a smallness of the integral increment norm $||\delta u||$ for decreasing $J_G$. The smallness $|\delta u|$ at the planning interval $\sigma = (T_o, T_f]$ is not necessary. The advantage of MSA is that it allows to implement needle control variations subject to the whole area of feasible control actions subject to the given constraint system, i.e., the area of feasible schedules. Another MSA advantage is that the search for an optimal control in each iteration is performed in the class of boundary (e.g., pointwise or relay) functions which correspond to the discrete nature of decision making in scheduling.

Note that the MSA in its initial form has not guaranteed the convergence. By now, a number of MSA modifications with proved convergence exist (Lyubushin, 1979). The following MSA modification can be used for the considered scheduling algorithm. In step 4, formula (31) is used for the maximization of the Hamiltonian:

$$
\begin{align*}
\tilde{u}^{(r+1)}(t) &= \begin{cases} 
    Ru^{(r)}(t), & t \in [t', t''] \\
    u^{(r)}(t), & t \notin [t', t''] 
\end{cases},
\end{align*}
$$

where $[t', t'']$ is selected so that a decrease

$$J_G^{(r+1)} < J_G^{(r)}
$$

is ensured.

$J_G^{(r)}$, $J_G^{(r+1)}$ are the values of the goal functional subject to $u^{(r)}$, $u^{(r+1)}$, respectively. In Eq. (31), the operator $R$ corresponds to step 4 of the DYN algorithm. The selection of $t', t''$ in each iteration is based on the values of the Hamiltonian functions of two subsequent iterations. An example is provided in Figure 7.

![Figure 7 Hamiltonian function values in different iterations](image)

At the Y-axis, the values of the Hamiltonian function (19) are presented. The X-axis represents the instants of the time interval [0, 29]. The three lines correspond to three
computational iterations and depict the values of the Hamiltonian function (19) at different time instants. The red line (starting at t=0 with the value 500) refers to the first iteration; the blue line (starting at t=0 with the value 550) refers to the second iteration; and the green line (starting at t=0 with the value 650) refers to the third iteration. It can be observed that within \( t \in [0,12] \), the schedule calculated at the third iteration provides better results. For instance, \( H_1 \) (Eq. 20) increases in the case of lower penalties. Similarly, \( H_2 \) (Eq. 21) increases in the case of higher values of \( \mu_{jiu} \) which means a higher volume of processed goods.

However, within \( t \in [12,27] \), the first scheduling iteration outperforms the other iterations. For \( t \in [27,29] \), the third iteration does not improve the results of the second iteration, i.e., the schedule of the second iteration for \( t \in [27,29] \) shall be used. In combining the upper bounds of the Hamiltonian functions, an optimal schedule is calculated. It contains, for this example, the results of the third iteration (green line) for \( t \in [0,12] \), the first iteration (red line) for \( t \in [12,27] \), and the second iteration (blue line) for \( t \in [27,29] \).

The dynamic coordination parameters are conjunctive variables which change their values during the iterative process of the corresponding two-point boundary problem solution. At each time instant, the global objective function is the Hamiltonian (19). The locally coordinated sub-problems are partial combinatorial assignment problems (Eqs. (20), (2)–(4)) and LP problems (Eqs. (21), (14)–(15)) which are formed dynamically in dependence on the current active scheduling zone (see Figures 2–5).

If so, the values of the conjunctive variables (i.e., the coordinated signals) change from one iteration to the next iteration and can be considered in some sense as a generalization of the coordination parameters previously considered in the decomposition–coordination procedures of Danzig and Wolfe (1960) (goal coordination) and Kornai and Lipták (1965) (resource coordination). In the proposed approach, the conjunctive variables and the Hamiltonian function allow at each time instant to perform both the goal and the resource coordination of the solutions which are found in the combinatorial sub-problems.

In addition, the developed modification of the MSA method guarantees a monotonic change of the conjunctive variable values by both the transversality conditions (28)–(30) and a situational selection of the Hamiltonian function values. The transversality conditions interconnect the state parameter values in the main and conjunctive systems (see Eqs. (1), (11), (13), (25)–(27)) at the time moment that corresponds to the end of the planning interval. The values of the Hamiltonian function (19) are saved during the iterative search procedure (see Figure 7). The theoretical convergence of the considered iterative procedure has been previously proved in Lyubushin (1979) and Chernousko and Lyubushin (1982).

6 Optimality and complexity analysis

In this section, we analyze the model and the algorithm from Sections 4 and 5.

6.1 Optimality and existence analysis

Proposition 1 The optimal program control \( u(t) \) is an optimal flow shop schedule.
Existence proof. The formulated model is a linear non-stationary finite-dimensional controlled system of differential equations with a convex area of feasible control. This model form satisfies the conditions of the existence theorem in Lee and Markus (1967, Theorem 4, Corollary 2), which allows us to assert the existence of an optimal solution in the appropriate class of feasible controls. According to Lee and Markus (1967, Theorem 4, Corollary 2), along with the initial class $\tilde{K}$ formed via the constraints $q^{(1)}$ and $q^{(2)}$ describing the domain $Q(x(t))$, an extended class $\tilde{K}$ of control inputs can be considered. In the extended class $\tilde{K}$, the relay constraints $u_{iij}(t) \in [0;1]$ are substituted by a less strict one $u_{iij}(t) \in [0;1]$ (u is substituted by $\tilde{u}$). In this case, an extended domain $\tilde{Q}(x(t))$ of feasible control inputs may be formed by means of special transformations ensuring the convexity and the compactness of $Q(x(t))$ (Moiseev, 1974). The theorem of Lee and Markus (1967) confirms that all the conditions for the existence of an optimal control for the extended control class $\tilde{K}$ are valid.

If in a given class of feasible control actions $\tilde{K}$, an optimal control $\tilde{u}(t)$ exists, then, as arises from the local cut method, the control $\tilde{u}(t)$ returns at each time instant $t \in [T_0, T_f]$ at the set $\tilde{Q}(x(t))$ the maximum to the Hamiltonian (19)–(21). The proof is complete.

Optimality proof. An analysis of (19)–(21) shows that the Hamiltonian is linear in $\tilde{u}$. Since $\tilde{Q}(x(t))$ is a linear capsule of $Q(x(t))$, the maximization of the Hamiltonian (19)–(21) over the sets $Q$ and $\tilde{Q}$ leads to the same results. An optimal control for the class $\tilde{K}$ belongs to the class $\tilde{K}$. Taking into account $\tilde{K} \subset \tilde{K}$, this control is also optimal for the class $\tilde{K}$. Therefore, the relaxed problem can be solved instead of the initial one to obtain an optimal feasible control for the class $\tilde{K}$. The proof is complete.

Corollary 2 An analysis of the studies (Boltyanskiy, 1973; Moiseev, 1974) shows that for a linear non-stationary finite-dimensional system (model $M$) with a convex area of a feasible control $Q(x)$ and the goal vector, the stated necessary optimality conditions are also sufficient conditions.

6.2 Analysis of the complexity of the algorithm

It is known from the literature that the MSA method can be easily implemented when programming an algorithm. It also allows a fast computation. One MSA iteration corresponds to one iteration of a gradient method. The MSA differs from the gradient methods in calculating the Hamiltonian function (19) in each iteration with the usage of the previous iteration data.

Proposition 2 The computational complexity of the MSA in one iteration is polynomial. With regard to the considered scheduling problem, the complexity of the proposed algorithm in one iteration is determined by the complexity of the IP assignment problem (20), (2)–(4) and the LP problem (21), (14)–(15).
Proof On the basis of the works on the Hungarian method, the complexity of the IP problem can be estimated according to the formula

\[ O(\overline{m}_i^3 \cdot \frac{\sigma}{\Delta t}), \]  

(33)

where \( O(\overline{m}_i) = \frac{11\overline{m}_i^3 + 12\overline{m}_i^2 + 3\overline{m}_i}{6} \) is the maximal number of the “+” operations at one time point of the planning period \( (T_o, T_f) \) for the assignment problem being solved via the Hungarian method;

\( \overline{m}_i \) is the maximal dimensionality of the assignment problem, i.e., the number of independent paths in the schedule execution network diagram. For example, in Figures 2–5, the dimensionality is equal to the job number \( \overline{B}^{(i)} \), i.e., \( \overline{n} = 6 \);

\( \sigma \) is the duration of the planning interval;

\( \Delta t \) is the step length of integration for the main and the conjugate system.

Note that the integration of the differential equations in the main and the conjugate system is not necessary due to their linearity. The computation can be performed subject to simple recurrent formulas. The step for the recurrent calculations may be variable, subject to significant events which influence the current («active“) field of operations (gray colored operations in Figures 2–5). These events may include, e.g., the completion of one operation, a pre-emption of an operation, or an arrival of a new operation.

Complexity of the LP problem (21), (14)–(15) can be estimated according to the complexity of the simplex method which can be used to solve the above-mentioned LP problem. This complexity can be estimated for one iteration as \( O((\overline{n} \cdot \overline{s} \cdot n)^2 + \overline{n} \cdot \overline{s} \cdot n^2) \), where \( \overline{n} \) is the maximal number of jobs executed in parallel within the planning interval, \( \overline{s} \) is the average operation number of the jobs, and \( n \) is the maximal number of machines working in parallel. Similar to (33), the complexity of the LP calculation within the MSA can be estimated as follows:

\[ O((\overline{n} \cdot \overline{s} \cdot n)^2 + \overline{n} \cdot \overline{s} \cdot n^2) \cdot \frac{\sigma}{\Delta t}, \]  

(34)

Thus, the total complexity at one MSA iteration is as follows:

\[ \overline{N} < [O(\overline{m}_i) + O((\overline{n} \cdot \overline{s} \cdot n)^2 + \overline{n} \cdot \overline{s} \cdot n^2)] \cdot \frac{\sigma}{\Delta t}, \]  

(35)

where \( \overline{N} \) is the average number of addition operations.

Since the complexity of the IP/LP problem at each cut is polynomial and the number of integration steps and iterations increases linearly, the computational complexity of the DYN algorithm is polynomial. The proof is complete.

7 Conclusions
In this paper, we considered non-deterministic issues in flow-shop scheduling where scheduling is interconnected to the control function. In an integrated manner, we included such factors like temporal unavailability of machines, pre-emption, and fluctuations of processing times and technological constraints. This forms a dynamic scheduling environment which is important in managerial practice.

In this study, an original approach to a dynamic decomposition of an NP-hard combinatorial scheduling problem has been presented. The decomposition is based on the developed model and an algorithm for the optimal control of the execution of the operations blended with MP. The proposed dynamic decomposition is supported both with an algorithm of local coordination with the help of MP (i.e., at each time instant) and an algorithm of global optimization (i.e., upon the whole planning horizon). This results in the formulation and solution of partial combinatorial problems of lower dimensionality.

In light of this result, the theoretical contribution of this study is directed towards increasing the scheduling quality with the help of a sophisticated scientific methodology. The proposed novelty of this study consists of a detailed theoretical analysis of the time-based decomposition and computational complexity with an application to flow-shop scheduling with continuous flows and discrete assignments. A dynamic model and an algorithm have been developed for the simultaneous solution of the assignment and flow control tasks.

Along with some demonstrated advantages, this approach is not developed in order to outperform the heuristics or MP algorithms but rather extends them in the following aspects. The results gained may be naturally included into general iterative search procedures for optimal programs (e.g., schedules). The results of this study suggest the following possible extensions to existing scheduling models: dynamics of the execution of the operations (i.e., control of non-stationary increasing processing flow); machine non-stationarity; and using continuous variables for scheduling continuous flows (e.g., petrochemistry, energy, oil and gas). From a practical point of view, ipso facto that a schedule and the execution control can be formulated as an OPC is a great advantage in order to enlarge the scope of SC analysis by obtaining analytical solutions or robustness properties and investigating different adaptation policies.

The main idea of the proposed modification of the classical OPC model is to implement and update (e.g., due to dynamic changes in capacity availability) non-linear constraints on a convex domain of feasible control inputs rather than in the right-hand sides of differential equations. In this case, the coefficients of the conjunctive system (i.e., the dynamic Lagrange coefficients), keeping the information about the operational and logical constraints, can be explicitly defined via the local cut method (Boltyanskiy, 1973).

Furthermore, we proposed to substitute the relay constraints by interval ones, i.e., instead of the relay constraints \( u_{ij} \in \{0,1\} \) less strict ones \( u_{ij} \in [0,1] \) can be considered. Nevertheless, the control takes Boolean values as it is caused by the linearity of the differential equations and the convexity. The proposed substitution enables us to use fundamental scientific results of the OPC theory in scheduling.

The formulated model is a linear non-stationary finite–dimensional controlled system of differential equations with a convex area of feasible control. This is the essential structural
property of the proposed approach, which allows to apply methods of discrete optimization for the OPC calculation and ensuring the required consistency between OPC and LP/integer programming (IP) models. Although the solver works in the space of piecewise continuous functions, the control actions (i.e., the assignments) can be presented in a discrete form as in LP/IP models.

The continuous time representation allows to analyse the execution of the operations at each time point, and therefore, to obtain additional information about the execution of the operations and the flow control (see Figure 6). Since the DYN algorithm is launched by heuristics procedures, the performance of these heuristics may be compared with each other and with the optimal DYN solution. The analysis showed that, since the complexity of the IP/LP problem at each cut is polynomial and the number of integration steps and iterations increase linearly, the computational complexity of the proposed DYN algorithm is also polynomial.

Among the limitations of this study, the strong orientation on centralized control and the lack of software tools for a comparative analysis with the existing benchmark solutions can be mentioned. This is the focus of our future efforts. As the convergence speed of the proposed algorithm depends on the selection of the heuristic solution to the vector of the conjunctive system, further research in this direction is needed, e.g., an application of higher-level heuristics.

References


Appendix 1. Notations

Sets
\( \overline{B} = \{B^i, i \in \overline{N}, \overline{N} = (1, \ldots, \overline{n}) \} \) is the set of jobs
\( D = \{D^\mu, \mu \in \overline{S}, \overline{S} = (1, \ldots, \overline{\mu}) \} \) is the set of operations
\( M = \{M^j, j \in N, N = (1, \ldots, n) \} \) is the set of machines
\( \Gamma \) is the set of precedence operations
\( Q(x(t)) \) is the domain of feasible control inputs
\( \tilde{Q}(x(t)) \) is the extended domain of feasible control inputs
\( \tilde{K} \) is the initial class of feasible control inputs
\( \tilde{\tilde{K}} \) is the extended class of feasible control inputs

Parameters
\( a \) is the planned processing volume
\( \tilde{R} \) is the total machine capacity
\( T_0 \) is the start instant of time of the scheduling horizon
\( T_f \) is the end instant of time of the scheduling horizon
\( c \) is the processing intensity
\( \xi(t) \) is the vector of perturbation impacts
\( \Delta t \) is the step length of integration for the main and the conjugate system.
\( h_0^{(o)}, h_1^{(o)} \) are known differentiable functions that determine the end conditions of the vector

Indices
\( i \) is the job index
\( j \) is the machine index
\( \mu \) is the operation index (i.e., number of the operation in the job)
\( l \) is the index of the flow-shop stage
\( o \) is the index of parameters and variables in the model \( M_o \)
\( k \) is the index of parameters and variables in the model \( M_k \)
\( f \) is the index of parameters and variables in the model \( M_f \)
\( z \) is the number of the Hamiltonian function
\( r \) is the number of the iteration of the algorithm

Continuous variables and functions
\( t \) is the current time instant
\( \sigma \) is the duration of the planning interval
\( x^{(o)} \) is a variable characterizing the state of an operation, where \( ^{(o)} \) indicates the relationship of the state variable \( x \) to the operation states.
\( \epsilon(t) \) is the given preset matrix time function of time-spatial constraints
\( \tilde{u}(t) \) is the decision control action at the moment \( t \) in the extended class \( \tilde{K} \)

\( J \) is a performance indicator

\( \alpha(\tau) \) is the penalty function in the mathematical model of the operation control processes

\( \beta(\tau) \) is the penalty function in the mathematical model for the flow control

\( x_j^{(k)} \) is a state variable characterizing the total employment time of machine \( M^{(j)} \)

\( x_j^{(r)} \) is a state variable characterizing the processed flow volume

\( u(t) \) is a feasible schedule

\( u^*(t) \) is an optimal schedule

\( H \) is the Hamiltonian function

\( \psi(t) \) is the conjunctive vector

\( q^{(1)} \) and \( q^{(2)} \) are vector-functions, defining the main spatio-temporal, economic, technical and technological conditions for the machine functioning process.

\( \delta_a(t), \rho_{\beta}(t) \) are coefficients of the conjunctive system

\( \lambda \) is the vector of the weight coefficients of the performance indicators

**Discrete variables**

\( u^{(o)}(t) \) is the decision control action at the moment \( t \)

\( \overline{m}_1 \) is the maximal dimensionality of the assignment problem

\( \tilde{n} \) is the maximal number of parallel executed jobs within the planning interval

**Others**

\( M_o \) is the mathematical model for the operation control processes (model)

\( M_k \) is the mathematical model for the machine control

\( M_f \) is the mathematical model for the flow control

\( M \) is the integrated model of the problem \( \text{PS} \)

\( \Lambda \) is the relaxed problem

\( R \) is an operator in the optimal control algorithm