HAL
archives-ouvertes.fr

A control approach to scheduling flexibly configurable jobs with dynamic structural-logical constraints

Dmitry Ivanov, Boris Sokolov, Weiwei Chen, Semyon Potryasaev, Alexandre Dolgui, Frank Werner

▶ **To cite this version:**

Dmitry Ivanov, Boris Sokolov, Weiwei Chen, Semyon Potryasaev, Alexandre Dolgui, et al.. A control approach to scheduling flexibly configurable jobs with dynamic structural-logical constraints. IISE Transactions, Taylor & Francis, 2021, 53 (1), pp.21-38. 10.1080/24725854.2020.1739787 . hal-02990899

HAL Id: hal-02990899
https://hal.archives-ouvertes.fr/hal-02990899

Submitted on 5 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Preprint

D. Ivanov, B. Sokolov, W. Chen, **A. Dolgui**, S. Potryasaev, F. Werner. A control approach to scheduling flexibly configurable jobs with dynamic structural-logical constraints, *IISE Transactions*, 2021, vol. 53, n° 1, p. 21–38, doi.org/10.1080/24725854.2020.1739787

A control approach to scheduling flexibly configurable jobs with dynamic structural-logical constraints

Abstract

This study theorizes how the optimal control paradigm can be applied to modeling and algorithmic solutions in manufacturing systems with dynamically changing hybrid structural-logical-terminal constraints. Our study conceptualizes and operationalizes a unique class of problems with both flexible machines and flexible jobs that can be frequently encountered in manufacturing systems with individualized products when process and schedule are created simultaneously. We offer a model to schedule jobs in manufacturing systems when the structural-logical terminal constraints are changing dynamically, and an algorithm to obtain a tractable solution analytically within the proven axiomatic of the optimal program control and mathematical optimization. We develop a dynamic decomposition methodology for modeling and control of schedules in highly flexible production systems combining the advantages of continuous and discrete optimization. The findings suggest that our approach can be of value for approaching problems with a simultaneous process design (i.e., task composition) and operation sequencing (i.e., service composition). Utilizing the outcomes of this research could also support the consideration of dynamics in the operations control. The operations execution can be accurately modeled in continuous time as state variables the updates of which allow for data-driven control of machine availability and capacity disturbances. Besides, the method developed theorizes further generalized insights into decomposition methods for scheduling and is supported by an analytical analysis and an algorithmic realization.

Keywords: manufacturing, dynamics, flexible machines, flexible jobs, optimal control

1. Introduction

Modelling and control in manufacturing have been frequently considered in light of flexible or reconfigurable systems, where multiple flexible machines are divided into modules to perform different tasks (Li and Meerkov 2009, Dolgui and Proth 2010). The machine flexibility allows for dynamically changing process technologies (i.e., flexibly arising jobs through customer-specific sequencing of operations) to achieve the product individualization and variety (Balakrishnan and Geunes 2003, Koren et al. 2016). However, the issues of scheduling of flexibly arising jobs have not been explicitly treated in the domain of reconfigurable systems. In scheduling, most of the problems presume a fixed processing technology for product manufacturing (Pinedo 2018). At the same time, the recent literature provides a number of insights on the scheduling with flexible jobs, especially in relation to digital manufacturing (Ivanov et al. 2016, Ahn et al. 2018, Liu et al. 2018).

Our study aims at integrating the two above-mentioned domains – flexible machines and flexible jobs – within a unified methodological framework. The recent literature (Dolgui et al. 2019) points to a special class of modeling and control problems with hybrid structural-logical-terminal constraints which are frequently encountered in customized manufacturing systems with individualized products when the process design and job sequencing are performed simultaneously (Battaia et al. 2018, Kusiak et al. 2018). The structural constraints usually relate to the process design while logical (i.e., precedence) constraints are concerned with job sequencing, and the terminal constraints are encountered in both process design and scheduling. One difficulty in modelling such problems is that those constraints are changing dynamically in time due to availabilities of machines and individual job structuring by operations.

Examples of such systems stem from new digital manufacturing paradigms such as Industry 4.0 and cloud manufacturing which bring new challenges to the scheduling community. In digital manufacturing, the process planning and job scheduling are integrated due to the usage of services instead of the rigid and fixed allocations of operations and machines (Frazzon et al. 2018, Rossit et al. 2018). These new technologies are enabling a highly flexible production, particularly through the use of cyber-phys-

ical systems and customized assemblies in order to deliver manufacturing services on-demand to consumers (Nayak et al. 2016, Ahn et al. 2018, Battaïa et al. 2018, Frazzon et al. 2018, Tao et al. 2018, Yang et al. 2019).

We build on and contribute to the literature on manufacturing systems with flexible machines and flexible jobs. The studies on reconfigurable manufacturing systems (Youssef and ElMaraghy 2006, Bortolini et al. 2018) and multi-stage, flexible, job-flow shop scheduling problems with flexible machines (Kyparisis and Koulamas 2006, Ivanov et al. 2016a,b, Bozek and Werner 2017) are the closest research streams to this new emerging scheduling area in the era of Industry 4.0. Another relevant research field for our study is the resource-constrained project scheduling with alternative activity chains (Tao and Dong 2017, Blazewicz et al. 2019). While providing the profound results for problems with a combination of either structural-terminal constraints (Kreter et al. 2016), or structural-logical constraints (Tao and Dong 2017), or structural-logical constraints (Battaïa et al. 2017), none of these research fields, however, comprehensively treats the combination of both flexible machines, and flexible jobs, as well as flexible dynamic structural-logical-terminal constraints – a substantive and distinctive contribution made by our study.

Many problems known in those and other relevant domains are already NP-complete as decision problems, and pose challenges to the mathematical programming and optimization communities even for fixed processes (Kyparis and Koulamas 2006, Tahar et al. 2006, Jungwattanakit et al. 2009, Tran and Ng 2011, Vallada et al. 2017). If the process fixation would now be released as posited in Industry 4.0 and cloud manufacturing, those problems would become even harder to formulate and to solve both in decision and in computational spaces. The traditional paradigm of tackling such a problem is to solve a mathematical program in the planning stage, and resolve the problem with updated information in scheduling. However, under the flexible manufacturing setup, the frequency of an information update is typically too fast to respond by solving mathematical programming models. The process flexibility may also render the solution obtained in the planning stage meaningless. As such, modeling and solution methods for the simultaneous structural-functional synthesis of the process design and scheduling in systems with decentralized computational activities and dynamically changing resource availabilities need to be further developed using a different paradigm.

Our study brings the discussion further and explores the domain of scheduling with dynamically changing structural-terminal-logical constraints by framing the decision-making problem and the decomposition technique in terms of optimal control blended with discrete optimization. We believe that the control theory can propose interesting frameworks to design scheduling problems when utilizing its combinations with local search or other traditional combinatorial optimization approaches. This is to say that the control paradigms can be useful to design frameworks in manufacturing scheduling in an integration with discrete optimization scheduling techniques.

There are several argumentations to support the necessity to explore new optimization techniques for scheduling in customized manufacturing systems looking at the interface of control and discrete optimization. First, scheduling in digital manufacturing exhibits a specific characteristic, i.e., a simultaneous synthesis of the processing technology (planning) and scheduling. As such, scheduling is now considered as a combination of process (task) composition and service composition which happen simultaneously, so the process and schedule are created at the same time. The known solutions in discrete optimization based on mathematical decomposition or relaxation have tackled some of these problems (e.g., Kaskavelis and Caramanis, 1998), but others remain challenging. More recent decomposition ideas, such as data-driven or clustering approaches (Chen et al. 2013a,b), are founded on the difficulties in deriving analytical properties. Metaheuristics or heuristics have also been widely used in solving these problems, such as job shop scheduling (Barnes and Chambers 1995; Lobo et. al. 2014), but no solution-quality guarantee can be provided. Second, a peculiarity of scheduling in digital manufacturing is its data-driven nature that, at the same time, becomes favorable for the development of new dynamic approaches to modelling those problems based on decomposition methods (Ivanov et al. 2016a, Rossit et al. 2018, Xu et al. 2019). We are not aware of any published formulation of those problems using mathematical programs which consider the dynamically changing structural-terminal-logical constraints. Even if such a new problem could be formulated as a mathematical program by finding an innovative form to describe the dynamically changing constraints/flexibilities as equations/inequalities, it is difficult to imagine an efficient solution technique as for now. The main peculiarity is that both the manufacturing process structures and the resource/machine availabilities are changing in time.

In summary, our study makes the following contributions to the literature. First and principally, we offer a model to describe the manufacturing scheduling problem with dynamically changing structural-terminal-logical constraints and an algorithm to obtain a tractable solution analytically within the proven axiomatic of the optimal control and mathematical optimization. We build on and contribute to the literature on scheduling in highly flexible and digital manufacturing, decomposition principles in flexible shop scheduling, and scheduling by optimal control. We consider a unique combination of both flexible machines, and flexible jobs, and flexible dynamic structural-logical-terminal constraints. With all that, our study conceptualizes and operationalizes a unique class of manufacturing control problems when the process and schedule are created simultaneously and the hybrid structural-terminal-logical constraints are subject to dynamic changes in time.

The remainder of this paper is as follows. Sect. 2 conducts a survey on the related literature and summarizes the main contributions of this paper. In Sect. 3, the problem under study is introduced, and the optimal control model is presented. Sect. 4 develops the computational procedure for solving the model, and illustrate the algorithm using an example. Finally, Sect. 5 concludes the paper by summarizing the most important results of this study and outlining some issues for future research.

2. State of the Art

The theoretical foundations of our approach build on four theoretical perspectives, namely reconfigurable manufacturing systems, flexible job scheduling, scheduling in digital manufacturing, and resource-constrained project scheduling. While the first and the latter ones can be considered as conceptual enablers of our study, the second and third perspectives have a closer relationship to the technical implementation. As such, and in light of the limited length of the paper, we restrict ourselves to their analysis.

A growing body of research has reported advances in dealing with flexible flow shop scheduling. 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 algorithm for this strongly NP-hard problem. Tazar et al. (2006) considered a scheduling problem for 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 (i.e., the makespan) is minimized. For this NP-hard problem, the study developed a heuristic algorithm using linear programming (LP). Furthering these

insights, Bożek and Werner (2018) developed an optimization method for flexible job shop scheduling with lot streaming and sublot size optimization. The recent literature has included a variety of principles and approaches to the design and scheduling of flexible and reconfigurable assembly systems with a focus on balancing, scheduling, and sequencing (Delorme et al. 2012, Battaïa et al. 2017). In these studies, models and methods have been presented for solving problems related to the optimization of the performance intensity of an assembly system for sets of flexibly intersecting operations. In particular, in scheduling with *alternative parallel machines* at each stage of the manufacturing process, alternative machines may execute the operations. Some research reported associations between the flexibility in the process plan and an increased complexity in both the machine assignment and sequencing of tasks (Yu et al. 2011, Blazewicz et al. 2015, Chen et al. 2018). An extensive body of the literature has been developed on simultaneously selected process plans and schedules as well as on scheduling with changing availability of resources, predominantly rooted in discrete optimization techniques. One can refer to the frameworks of multi-processor flow shops, hybrid flow tshops with identical machines, unrelated parallel machine problems with machine and job selection, distributed assembly permutation flow shops, multi-objective hybrid flow shop problems, and complex hybrid flexible flowline problems (Jungwattanakit et al. 2009, Tran and Ng 2011, Fanjul and Ruiz 2012, Vallada et al. 2017).

The second perspective directs the attention to the main specifics of scheduling in digital manufacturing. The paper by Ivanov et al. (2016a) was among the first studies conducted on the impact of Industry 4.0 on manufacturing scheduling. It proposed a model and algorithm based on a combination of continuous and discrete optimization that allows for representing schedules in the form of optimal control trajectories which are dynamically driven by data transformed to state and control variables. Frazzon et al. (2018) underlined the role of digitalization to apply data-driven adaptive planning and control. The data integration between the real manufacturing system and the simulation model was implemented through a data-driven framework. Ivanov et al. (2018) considered proactive scheduling and reactive real time control in Industry 4.0 manufacturing systems and developed a concept and model of data-driven schedule optimization. Rossit et al. (2018) developed a concept of smart scheduling, aiming to yield flexible and efficient production schedules that are formed dynamically, in an event-oriented matter. In smart manufacturing, the processes for different customer orders may have individual structures of the stations

such that the flexible stations are able to execute different functions subject to individual sets of operations within the jobs (Ivanov et al. 2016a,b, Nayak et al. 2016, Battaïa et al. 2017, Zhong et al. 2017). Such a perspective on the integrated problem of simultaneous, structural-functional synthesis of the Industry 4.0 customized assembly systems and job scheduling appears to be more relevant in a practical decision environment, relative to inherent problems in incorporating changes in the environment and production system itself.

Similar approaches have been developed in the area of cloud manufacturing, the concept of which aims to deliver manufacturing services to consumers on-demand. Liu et al. (2018) pointed out task composition and service composition as two dominants in scheduling in cloud manufacturing. Task composition is a prerequisite for scheduling of tasks in cloud manufacturing. Service composition is a combination of multiple services, atomic or composite, into value-added services to fulfil a task or a set of tasks. Ahn et al. (2018) analyzed the selection of services from the cloud manufacturing platform and combining them to a process by task composition.

The analysis of the concept of services leads to two fundamental specifics of digital manufacturing. The first one is the production system redundancy, and the second one is the computational redundancy. The production system redundancy is driven by industrial Internet-of-Things (IoT) which transforms production processes from rigid links between operations and machines to a variety of flexible allocations. The computational redundancy arises due to the furnishing of the machines and products with computational and data processing devices, allowing performing some computations directly by machines and products and using Cloud and Fog computational technologies. These two redundancies create a principally new entity for manufacturing modelling and control, i.e., a service. The services can realize almost any combination of machines and operations (Ivanov et al. 2016a, Ahn et al. 2018, Liu et al. 2018). None of these studies, though, formally and rigorously modelled the manufacturing systems with mixed structural-terminal-logical constraints and considering the data-driven schedule updates – a distinctive and significant contribution made by our study.

In summary, the studies on scheduling in Industry 4.0, smart manufacturing, and cloud manufacturing differ regarding the methodologies but most of them share a certain set of the following attributes: (a)

simultaneous structural-functional synthesis of the customized assembly system, (b) data-driven scheduling based on the communication of the products and stations with a resulting reduction of the combinatorial complexity, and (c) resemblance to flexible multi-processor flow shop scheduling. For example, in the manufacturing platform MindSphere which is a cloud-based, open IoT operating system, products, plants, systems, and machines are connected to enable the usage of data generated by the IoT with advanced analytics in schedule optimization (Siemens 2018). Usage of autonomous mobile robots (AMR) makes it possible to implement almost any flexibility in the manufacturing processes even with the traditional assembly line stations (Nielsen et al. 2017). However, in previous studies, the selection of the process structure and the respective station functionality for the execution of the operations were considered in isolation. In many cloud manufacturing, such an integration can have a significant impact on the process efficiency (Liu et al. 2018). The problem of a simultaneous structural-functional synthesis of the customized assembly system is still at the beginning of its investigation. Previously isolated insights gained in hybrid shop scheduling and sequencing with alternative parallel machines, resource constrained project scheduling with alternative chains, reconfigurable manufacturing system design and optimal control scheduling models with both terminal and logical constraints can now be integrated into a unified framework of Industry 4.0 and require an extension towards models with hybrid structural-terminal-logical constraints (Dolgui et al. 2019). With that being said, doubts have been raised as to whether efficient solutions can be proposed to the scheduling problems with combined logical-structural constraints using traditional discrete optimization paradigms which directs our attention to continuous optimization and control theory.

While the automatic control community has developed a body of knowledge for scheduling of manufacturing, sensor systems and real-time scheduling of dynamic control systems (Burgess and Passino 1999, Sethi et al. 2000, Aicardi et al. 2008, Park and Yang 2009, Mati and Xie 2011, Vincent and Ponnambalam 2013, Han et al. 2017), the use of a control paradigm for manufacturing scheduling is still restricted despite of a proven resemblance of the manufacturing systems to control systems (Feng and Yan 2000, Ivanov et al. 2018, Dolgui et al. 2019, Sokolov et al. 2018). As a result, it is not yet clear how the control paradigm can be applied to scheduling in highly flexible systems of the digital manufacturing.

Our method combines the advantages of continuous and discrete optimization. Specifically, our study develops an optimal control model and algorithm for the simultaneous structural-functional design of a customized manufacturing process and the sequencing of the operations within the jobs. The developed theoretical framework presents a contribution to flexible scheduling in the emerging field of Industry 4.0-based innovative production systems. We mention that this study extends previous publications of the authors (Ivanov et al. 2016a,b), where only scheduling decisions were considered and the design of the process structure was assumed to be fixed, i.e., the design of a flow shop process was considered. In this paper, the scheduling model considers the structural synthesis and sequencing decisions of the manufacturing process, and therefore it forms a more complicated problem structure, where the scheduling control framework becomes critical in solving large-scale dynamic problems.

A distinctive feature of our approach is that we propose to decompose dynamically the large-scale assignment matrix according to the precedence relations between the operations of the jobs and dynamically consider only the operations that satisfy these precedence relations at a given time point in small-dimensional, discrete optimization models. Discrete optimization algorithms are used for scheduling in these matrices of small dimension at each time point. Continuous optimization algorithms (e.g., the method of successive approximations, or the method of Krylov-Chernousko, see Ivanov et al. 2016 and Dolgui et al. 2019, and the references in these papers) are used to create a schedule from the discrete optimization results generated at each time point by extremizing the Hamiltonian function at each time point subject to some criteria (e.g., tardiness). The present study applies the dynamic decomposition method to fully flexible flow shops with both alternative input and output precedence relations (cf. Fig. 3 later in the paper). Such a combination is unique in the literature, as it mimics the complexity of modern manufacturing reality affording more realistic applications.

The findings suggest that our approach can be of value for problems with a simultaneous process design (i.e., task composition) and operation sequencing (i.e., service composition). Utilizing the outcomes of this research can also help to accurately model the dynamics of the operations execution in continuous time as state variables the updates of which allow for a data-driven control of machine availability and capacity disturbances. The distinctive and significant contribution made by our study is that the con-

structed modelling paradigm (scheduling control framework, combining discrete optimization and optimal control) is explicitly capable of capturing dynamic features in digital manufacturing, which are difficult to model using the traditional mathematical programming paradigm. Such scenarios are challenging to model as a linear relationship in discrete optimization, but are convenient to describe using the continuous control paradigm to address the complexity of the dynamically changing hybrid structural-logical-terminal constraints by decomposition principles. We should point out that the proposed approach does not replace the mathematical modeling approaches. Instead, it supplements the existing approaches in tackling a set of problems with new requirements (i.e., dynamically changing hybrid structural-logical-terminal constraints) emerging in digital manufacturing.

3. Scheduling Control Modelling Methodology

In this section, we present the scheduling control modeling methodology.

3.1 Problem Description

We now introduce the problem to be solved. Specifically, consider a customer job control system which interacts with an assembly system while the latter offers services to solve particular tasks in the job control system ((Fig. 1).

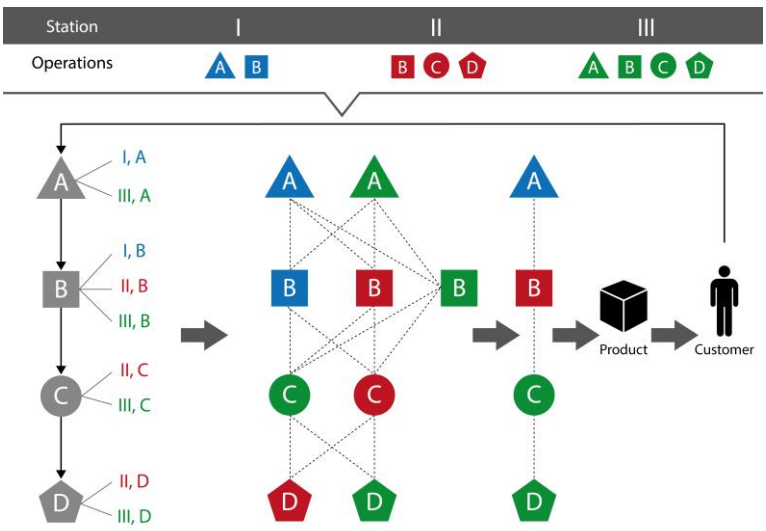


Fig. 1. Interaction of the customer and the assembly systems

The customer system generates orders (jobs) each of which has an individual sequence of the technological operations resulting in individual task composition. Each of these sequences requires an individual service composition. The interaction of the customer and the assembly systems results in alternatives

for the design of the manufacturing process. Each station has multiple channels, each of which is dedicated to a set of technological operations. Multiple stations may perform equal operations and alternatives for job scheduling and sequencing exist resulting in dynamic logical constraints which are, in turn subject to actual capacity utilization, machine availability, and time- and cost-related parameters of the services which represent terminal constraints. As such, there are dynamic structural constraints on the process design.

In Fig. 1, a customer order (i.e., a job) is considered which contains the four operations A, B, C, and D. Three flexible workstations are considered. Station I is capable of processing operations A and B, station II is capable of processing operations B, C, and D, and station III is capable of processing all operations A, B, C, and D. The first task is to design the manufacturing process, i.e., to perform a task composition by combining technological operations into a manufacturing process. In other words, the sequencing of operations into a process is performed. The second task is to implement a service composition by assigning the operations to stations at each stage of the technological process.

Such a problem shows a set of special features. For example, the dynamically changing operations sequences in the tasks may impose dynamic and nonlinear requirements in the constraints for the machine and operations control. Machine maintenance requirements and schedules can depend on its assigned tasks and sequence of operations, and thus the machine availability and capacity may change dynamically as a result, sometimes in a nonlinear relationship with time. Such dynamic and nonlinear constraints could be hard to model and capture in the traditional mathematical programming paradigm. In the next section, we propose a scheduling model for the given problem class with hybrid structural-functional-logical constrains which is based on a dynamic interpretation of the operations execution processes.

3.2 Model

Let us introduce the following basic sets and structures (the indices (O), (Π) and (P) describe the relations of the sets to the operations, flows, and resources, respectively).

$A = \{A_i, i = 1, \dots, n\}$ is the set of customer orders (jobs).

$M = \{M_j, j = 1, \dots, m\}$ is the set of resources (e.g., stations or machines; these terms are used as synonyms in the model description) in the manufacturing system.

$D^{(i)} = \left\{D_\kappa^{(i)}, \kappa = 1, \dots, S_i\right\}$ is the set of operations which can be executed in the system.

$P = \left\{P_\kappa^{(i)}, \kappa = 1, \dots, S_i\right\}$ is the set of material, energy or information flows in the process.

An intersection of the sets *M*, *D*, and *P* is a *service*; i.e., an entity needed to fulfill a process step in the set A. The services are formed dynamically based on the available machines and operations.

Assume that the manufacturing and transportation capacities may be disrupted and

- the station availability is described by a given preset matrix time function $\varepsilon_{ij}(t)$ of time-spatial constraints: we have $\varepsilon_{ij}(t) = 1$, if the station is available and $\varepsilon_{ij}(t) = 0$, otherwise;
- the station ability to process operations can be described by the set Θ whose element $\Theta_{i\kappa j}$ is equal to 1 if station M_j has the ability to process an operation $D_\kappa^{(i)}$, and equals 0, otherwise;
- the capacity dynamics can be described by a continuous function of the perturbation impacts $\xi_j(t)$; $\xi_j(t) = 1$ if 100% available, $\xi_j(t) = 0$ if the capacity is fully disrupted, and $\xi_j(t)$ = [0;1] in the case of partial unavailability.

The formal statement of the scheduling problem is based on a dynamic interpretation of the execution processes of the services. Let us introduce some new notations.

Parameters

$a_{i\kappa}^{(\mathrm{O})}$, $a_{i\kappa}^{(\Pi)}$ are the normative and actual flow times of each operation. The values of these parameters are related to the end conditions to be reached in $x_{i\kappa}^{(\mathrm{O})}$,$x_{i\kappa}^{(\Pi)}$ at $t = t_f$. While the normative time $a_{i\kappa}^{(\mathrm{O})}$ is fixed and known in advance, the actual operation execution time $a_{i\kappa}^{(\Pi)}$is computed during the optimization procedure by varying the processing intensities at the machines subject to the objective function extremization.

$c_{i\kappa j}^{(\Pi)}$ is the maximum processing intensity for the operation $D_\kappa^{(i)}$ at the M_j; it determines the maximum possible value for the production quantity for the operation $D_\kappa^{(i)}$ described by the control variable $u_{i\kappa j}^{(\Pi)}$.

μ_j is the manufacturing costs per time unit at M_j

Decision variables

$x_{i\kappa}^{(\mathrm{O})}$ is a state variable characterizing the normative flow time for operation $D_\kappa^{(i)}$ at each moment t;

$x_{i\kappa}^{(\Pi)}$ is a state variable characterizing the actual flow time of the operation $D_{\kappa}^{(i)}$ at each moment t; $x_{i\kappa}^{(\Pi)}$ can be varied (i.e., expedited and slowed down) according to variation of $c_{i\kappa j}^{(\Pi)}$

$x_j^{(\mathrm{P})}$ is a state variable characterizing the run time of the station M_j;

$u_{i\kappa j}^{(\mathrm{O})}(t)$, $\omega_{i\kappa j}^{(\mathrm{O})}(t)$ are control variables.

The meaning of the auxiliary (marked with index *B*) variables

$v, \omega^{(\mathrm{O})}, u^{(B,1)}, u^{(B,2)}, z, h, g, x^{(B,1)}, x^{(B,2)}$ is explained below and shown in Fig. 2.

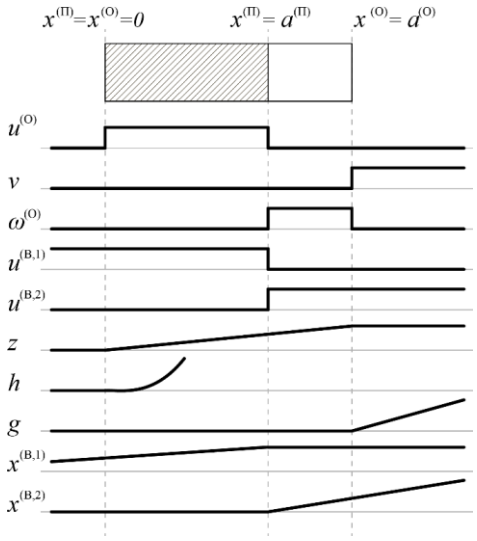


Fig. 2. Auxiliary control variable dynamics

z is an auxiliary variable which characterizes the execution of the *k*-operation, h is the square under the integral curve z, and g is an auxiliary variable which is equal to the time point *t* between the completion time of the *k*-operation and t_f. The meaning of the other auxiliary variables will be explained in the context of the constraint system (7)-(20). These auxiliary variables are used to fulfill the non-preemption condition for operations. The modeling procedure for the non-preemption of operations is shown in (Ivanov et al. 2013 and Sokolov et al. 2018).

Let us illustrate the general logical-dynamic mathematical model of the manufacturing process design and control. The problem statement described above implies both the process design and operation sequencing. As such, scheduling problems with *hybrid structural-terminal-logical constraints* must be analyzed (Dolgui et al. 2019). Consider the simplified example of a system with *hybrid structural-terminal-logical constraints* in Fig. 3.

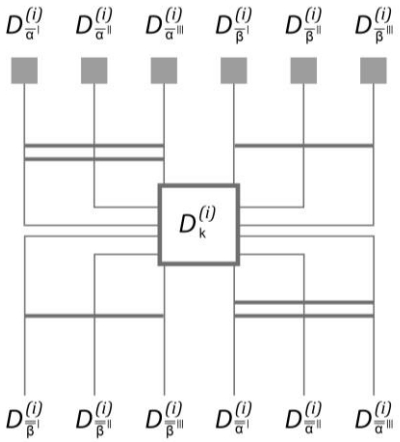


Fig. 3. Precedence relations of operation $D_{\kappa}^{(i)}$

In Fig. 3, the operation $D_{\kappa}^{(i)}$ follows the operations $D_{\tilde{\alpha}'}^{(i)} D_{\tilde{\alpha}''}^{(i)} D_{\tilde{\alpha}'''}^{(i)}$ according to the "AND" precedence rule (meaning that all the operations$D_{\tilde{\alpha}'}^{(i)} D_{\tilde{\alpha}''}^{(i)} D_{\tilde{\alpha}'''}^{(i)}$ should be completed to start $D_{\kappa}^{(i)}$; those operations are noted by index α) and the operations $D_{\tilde{\beta}'}^{(i)} D_{\tilde{\beta}''}^{(i)} D_{\tilde{\beta}'''}^{(i)}$ according to the precedence rule "OR" (meaning that at least one of the operations $D_{\tilde{\beta}'}^{(i)} D_{\tilde{\beta}''}^{(i)} D_{\tilde{\beta}'''}^{(i)}$ should be completed to start $D_{\kappa}^{(i)}$; those operations are noted by index β). Analogously, the six operations follow the operation $D_{\kappa}^{(i)}$ according to either "and" or "or" rules. The combination of "AND" and "OR" precedence and follower relations results in dynamic logical constraints which represent the highly flexible set of available technological process designs (task composition) and service compositions.

For the system considered, the optimal control model can be written as Eqs. (1)-(3).

$$\dot{x}_{i\kappa}^{(\mathrm{O})} = \sum_{j=1}^{m} \varepsilon_{i\kappa j}(t) \cdot \Theta_{i\kappa j} \left(u_{i\kappa j}^{(\mathrm{O})}(t) + \omega_{i\kappa j}^{(\mathrm{O})}(t) \right), \forall i, \kappa \tag{1}$$

$$\dot{x}_{i\kappa}^{(\Pi)} = \sum_{j=1}^{m} u_{i\kappa j}^{(\Pi)}(t), \forall i, \kappa \tag{2}$$

$$\dot{x}_{j}^{(\mathrm{P})} = \sum_{i=1}^{n} \sum_{\kappa=1}^{S_i} u_{i\kappa j}^{(\mathrm{O})}(t), \forall j \tag{3}$$

Eq. (1) describes the dynamics of the operation execution for the job A_i. If $\dot{x}_{i\kappa}^{(\mathrm{O})} = \sum_{j=1}^{m} u_{i\kappa j}^{(\mathrm{O})}(t)$, then at each time point, if $u_{i\kappa j}^{(\mathrm{O})}(t) = 1$ (i.e., the volume of the state variable $x_{i\kappa}^{(\mathrm{O})}$ is increasing), the job pro-

cessing is in progress at the M_j. Eq. (1) can be interpreted as a resource project scheduling model describing the network of operations. Eq. (2) describes the actual execution of the operations. Subject to the different possible operation assignments to the stations, $\dot{x}_{i\kappa}^{(\Pi)}$ can be expedited by accelerating the machine or selecting the fastest machine or slowed down. This processing progress is reflected by an increase in $u_{i\kappa j}^{(\Pi)}(t)$, e.g., with each additional output unit produced. Eq. (3) describes the dynamics of the station utilization. In case of $u_{i\kappa j}^{(\mathrm{O})}(t) = 0$, the station does not produce anything at time *t*. Similarly, if $u_{i\kappa j}^{(\Pi)}(t) = 0$, the product is not processed at time *t*.

Eqs. (1)-(3) are interconnected. While the process design (i.e., the technology synthesis) is described by Eq. (1), the sequencing is controlled by Eq. (2) by adjusting the flow time through the selection of processing intensities which allows to extremize the objective function. At the same time, the results of the process design directly affect the machine utilization using Eq. (3). In other words, Eqs. (1)-(3) describe the dynamic structural-functional synthesis of a flexible manufacturing system.

Eqs. (4)-(6) describe the requirements on the non-preemption of the operations and the numerical assessment of the number of completed operations as well as on the determination of the operation completion time, respectively.

$$\dot{z}_{i\kappa j} = u_{i\kappa j}^{(\mathrm{O})} + \omega_{i\kappa j}^{(\mathrm{O})}; \dot{h}_{i\kappa j} = z_{i\kappa j}; \dot{g}_{i\kappa j} = v_{i\kappa j}, \forall i, \kappa, j \tag{4}$$

$$\dot{x}_{i\kappa}^{(B,1)} = u_{i\kappa}^{(B,1)}, \forall i, \kappa \tag{5}$$

$$\dot{x}_{i\kappa}^{(B,2)} = u_{i\kappa}^{(B,2)}, \forall i, \kappa \tag{6}$$

The constraint system can be described as shown in Eqs. (7)-(20).

$$\sum_{j=1}^{m} u_{i\kappa j}^{(\Pi)}(t) \leq 1\,, \forall\, i, \kappa; \sum_{i=1}^{n} \sum_{\kappa=1}^{S_i} u_{i\kappa j}^{(\mathrm{O})}(t) \leq 1, \forall j \tag{7}$$

$$\sum_{j=1}^{m} u_{i\bar{\alpha} j}^{(\mathrm{O})} \cdot \sum_{\kappa \in \Gamma_{i\bar{\alpha}}^{+}} x_{i\kappa}^{(\mathrm{O})} = 0, \forall i, \bar{\alpha} \tag{8}$$

$$\sum_{j=1}^{m} u_{i\bar{\beta} j}^{(\mathrm{O})} \cdot \prod_{\kappa \in \Gamma_{i\bar{\beta}}^{+}} x_{i\kappa}^{(\mathrm{O})} = 0, \forall i, \bar{\beta} \tag{9}$$

$$\sum_{j=1}^{m} u_{i\bar{\tilde{\alpha}}j}^{(\mathrm{O})} \cdot \sum_{\kappa \in \Gamma_{i\bar{\tilde{\alpha}}}^{-}} \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)} \right) = 0, \forall i, \bar{\tilde{\alpha}} \tag{10}$$

$$\sum_{j=1}^{m} u_{i\bar{\tilde{\beta}}j}^{(\mathrm{O})} \cdot \prod_{\kappa \in \Gamma_{i\bar{\tilde{\beta}}}^{-}} \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)} \right) = 0, \forall i, \bar{\tilde{\beta}} \tag{11}$$

$$\sum_{j=1}^{m} \omega_{i\kappa j}^{(\mathrm{O})} \left[a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)} \right] = 0, \forall i, \kappa \tag{12}$$

$$\omega_{i\kappa j}^{(\mathrm{O})} \cdot g_{i\kappa j} = 0, \forall i, \kappa, j \tag{13}$$

$$v_{i\kappa j} \left[a_{i\kappa}^{(\mathrm{O})} - \sum_{j=1}^{m} z_{i\kappa j} \right] = 0, \forall i, \kappa, j \tag{14}$$

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{k}=1}^{\mathrm{s_i}} \mathrm{u}_{\mathrm{ikj}}^{(\Pi)}(\mathrm{t}) \leq \tilde{\tilde{\mathrm{R}}}_{\mathrm{j}}^{(\mathrm{f})} \cdot \xi^{(\mathrm{f})}(\mathrm{t}) \ , \forall i, \kappa, j \tag{15}$$

$$u_{i\kappa}^{(B,1)} \cdot x_{i\kappa}^{(B,2)} = 0, \forall i, \kappa \tag{16}$$

$$u_{i\kappa}^{(B,2)} \left[a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)} \right] = 0, \forall i, \kappa \tag{17}$$

$$0 \leq \xi_j(t) \leq 1, \forall j \tag{18}$$

$$0 \leq u_{i\kappa j}^{(\Pi)} \leq c_{i\kappa j}^{(\Pi)} u_{i\kappa j}^{(\mathrm{O})}(t) \xi_j(t), \forall i, \kappa, j \tag{19}$$

$$u_{i\kappa j}^{(\mathrm{O})}(t), \omega_{i\kappa j}^{(\mathrm{O})}(t), u_{i\kappa}^{(B,1)}(t), u_{i\kappa}^{(B,2)}(t) \in \{0,1\}, \forall i, \kappa, j \tag{20}$$

The structural constraints (7) determine the number of operations that can be simultaneously processed at a station. By default, this number is 1. The logical constraints are represented by Eqs. (8)-(12). Eqs. (8) and (9) define precedence relations for operation $D_{\kappa}^{(i)}$ with regard to the predecessor operations $D_{\tilde{\alpha}}^{(i)}$, $D_{\tilde{\beta}}^{(i)}$. Constraints (10) and (11) define precedence relations for operation $D_{\kappa}^{(i)}$ with regard to the subsequent operations $D_{\bar{\tilde{\alpha}}}^{(i)}$, $D_{\bar{\tilde{\beta}}}^{(i)}$. Constraint (12) defines the logic for the auxiliary control variable $\omega_{i\kappa j}^{(\mathrm{O})} \in \{0,1\}$ which equals 1 if $x_{i\kappa}^{(\Pi)}(t) = a_{i\kappa}^{(\Pi)}$ at time point t and $x_{i\kappa}^{(\mathrm{O})} \neq a_{i\kappa}^{(\mathrm{O})}$. In other words, the flow processing is completed. To compensate this, an auxiliary control $\omega_{i\kappa j}^{(\mathrm{O})}$ is introduced in Eq. (12). Constraints (13) and (14) are used to avoid overproduction regarding the operation $D_{\kappa}^{(i)}$, i.e., $x_{i\kappa}^{(\mathrm{O})}(t_f) = a_{i\kappa}^{(\mathrm{O})}$, which means that $x_{i\kappa}^{(\mathrm{O})}(t_f)$ cannot exceed $a_{i\kappa}^{(\mathrm{O})}$. The terminal constraints (15) determine the maximum capacity

$\tilde{\tilde{\mathrm{R}}}_{\mathrm{j}}^{(\mathrm{f})}$ of the station M_j. Eq. (15) is used jointly with constraints (13) and (14) to avoid interruptions of an operation execution (Ivanov et al. 2013). With the help of constraints (16) and (17), the jobs can be counted at $t = t_f$ for which all the required operations $D_\kappa^{(i)}$ are completed. In order to assess the schedule robustness (e.g., using attainable sets), we introduce a constraint on perturbation (18). Eq. (19) is a terminal capacity constraint on the operation processing subject to the maximum processing intensity $c_{i\kappa j}^{(\Pi)}$ at the stations. Since $c_{i\kappa j}^{(\Pi)}$ is different at various stations, the flow time of an operation is also variable depending on the station selection. With the help of Eq. (19), the state variables $x_{i\kappa}^{(\mathrm{O})}$ and $x_{i\kappa}^{(\Pi)}$ are connected. If $u_{i\kappa j}^{(\mathrm{O})}(t) = 1$, then we have a run of job $D_\kappa^{(i)}$ using M_j, and $u_{i\kappa j}^{(\mathrm{O})}(t) = 0$ otherwise; $\omega_{i\kappa j}^{(\mathrm{O})}(t) = 1$ at the moment when $x_{i\kappa}^{(\Pi)} = a_{i\kappa}^{(\Pi)}$, and $\omega_{i\kappa j}^{(\mathrm{O})}(t) = 0$ at the moment when $x_{i\kappa}^{(\mathrm{O})} = a_{i\kappa}^{(\mathrm{O})}$. Constraints (19) are used jointly with constraints (15). The latter determines the speed of the operations execution at the station while (19) determines the upper capacity limits at each station for all the operations executed at this station. Eq. (20) is a binary constraint.

Eqs. (8)-(12) can be considered as active dynamic constraints meaning that the number of operations in those constraints is changing dynamically in time in relation to Eqs. (1)-(3). The expressions in Eqs. (8)-(12) are to equal zero if, and only if control variables are equal to 1 which means that all the predecessor operations have been executed. This leads by tendency to the constraint system of a small dimensionality at each point of time that can be resolved using discrete optimization techniques for integer and linear programming. Along with the operations, flow and station dynamics in process control models (1)-(3), the dynamic changes in constraints (8)-(12) determine the dimensionality of the scheduling problem in a discrete optimization model at each point of time that consists of the constraints (17)-(20) subject to the criteria (23)-(24) and (26)-(28). Note that constraints (17)-(20) correspond to a canonical form of mathematical programming scheduling models.

The boundary conditions can be written as shown in Eqs. (21) and (22).

$$t = t_0 : x_{i\kappa}^{(\mathrm{O})}(t_o) = x_{i\kappa}^{(\Pi)}(t_o) = x_j^{(\mathrm{P})}(t_o) = z_{i\kappa j}(t_o) = u_{i\kappa j}(t_o) = g_{i\kappa j}(t_o) = 0$$
$$x_{i\kappa}^{(B,1)}(t_o) = 0; \quad x_{i\kappa}^{(B,2)}(t_o) = 0; \tag{21}$$

$$t = t_f : x_{i\kappa}^{(\mathrm{O})}(T_f) = a_{i\kappa}^{(\mathrm{O})}; \quad x_{i\kappa}^{(f)}(t_f) = a_{i\kappa}^{(f)} \tag{22}$$

$$x_i^{(\Pi)}(t_f), z_{i\kappa j}(t_f), h_{i\kappa j}(t_f), g_{i\kappa j}(t_f), x_{i\kappa}^{(B,1)}(t_f), x_{i\kappa}^{(B,2)}(t_f) \in \Re^1, \Re^1 \in [0, \dots, \infty)$$

The control functions are introduced in Eqs. (23)-(28).

$$J_1 = \frac{1}{2}\sum_{i=1}^{n}\sum_{\kappa=1}^{S_i}\left(a_{i\kappa}^{(O)} - x_{i\kappa}^{(O)}\right)^2 \to min \tag{23}$$

$$J_2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{\kappa=1}^{S_i}\left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right)^2 \to min \tag{24}$$

$$J_3 = \frac{1}{2}\sum_{i=1}^{n}\sum_{\kappa=1}^{S_i}\left[(a_{i\kappa}^{(O)} - \sum_{j=1}^{m} z_{i\kappa j})^2 + \sum_{j=1}^{m}(z_{i\kappa j} g_{i\kappa j} + \frac{(a_{i\kappa})^2}{2} - h_{i\kappa j})^2 z_{i\kappa j}^2\right]_{t=t_f} \to min \tag{25}$$

$$J_4 = \sum_{j=1}^{m}\left(x_j^{(P)}(t_f)\cdot \mu_j\right)^2 \to min \tag{26}$$

$$J_5 = \frac{1}{2}\sum_{\kappa=1}^{S_i}\left(x_{i\kappa}^{(B,1)}(t_f)\right)^2 \to min \tag{27}$$

$$J_6 = (J_1^{(min)} - J_1^{(\xi)})(J_2^{(min)} - J_2^{(\xi)}) \to min \tag{28}$$

Eqs. (23) and (24) are used to estimate the fullness of operation fulfillment subject to the planned volumes a_{ik}. Eq. (25) considers a non-preemptive execution of an operation. Eq. (26) reflects the costs incurred at $t = t_f$, where μ_j is the manufacturing cost per time unit at the M_j. Eq. (27) is used to estimate the fullfillment of operation i in time. The lower the value of J_5 , the shorter the makespan. Eq. (28) is used to estimate the schedule robustness using attainable sets (Ivanov and Sokolov 2010, Ivanov et al. 2016a,b) which enables a feasibility analysis of a two-point boundary problem.

4. Solution Algorithm

In this section, we present the solution algorithm that solves the model proposed in Section 3. Most centrally, we would like to point out that this study does not intend to enter a computational competition with the other known solution techniques for the studied problem but rather conceptualizes an entirely new viewpoint. This study theorizes how optimal control paradigms can be used to design a framework for modeling and optimal algorithmic solutions for scheduling problems with dynamically changing hybrid structural-logical-terminal constrains; one practical example of such problems is an integrated task and service composition in digital manufacturing scheduling.

The modelling and computational procedures are methodologically unified within the axiomatic of optimal control theory and based on using continuous state variables to determine the start and completion

times of the manufacturing service realization. The discrete control variables are used to compose services by assigning operations to the machines at different steps in the process. The execution of operations is described with the help of continuous state variables. The control variables of the operations are binary and discrete aiming at the assignment of operations to machines to form a service. The flow control variables are continuous and describe the actual volume of executed operations. The flow and operations control are interconnected through the logical-dynamic constraints (10)-(12).

4.1. Computational Principles

The major algorithmic idea proposed in this research is to decompose the assignment matrix dynamically in time, considering polynomially solvable (by tendency) problems of small dimensionality at each time point (see Fig. 4 for an example). In the example illustrated in Fig. 4, six jobs are considered, each of which contains a sequence of operations that can be executed at four alternative workstations. At time point $t=t_1$, the operations D_{62}, D_{51}, D_{41}, D_{31}, D_{21}, and D_{12} are available for an assignment to machines subject to the precedence relations. Therefore, a small-dimensional assignment problem with six operations and four machines arises at time point $t=t_1$ which is solved using standard discrete optimization techniques. As a result, the operations D_{62}, D_{51}, D_{31}, and D_{12} are assigned to the machines $M^{(4)}$, $M^{(3)}$, $M^{(2)}$, $M^{(1)}$, respectively. This is indicated by the control variables u_{624}, u_{513}, u_{312}, and u_{121}, which are switched to 1. Similarly, at time point $t=t_2$, we deal with a small-dimensional assignment problem with six operations and four machines. As a result, the operations D_{63}, D_{41}, D_{22}, and D_{13} are assigned to the machines $M^{(2)}$, $M^{(4)}$, $M^{(1)}$, $M^{(3)}$, respectively.

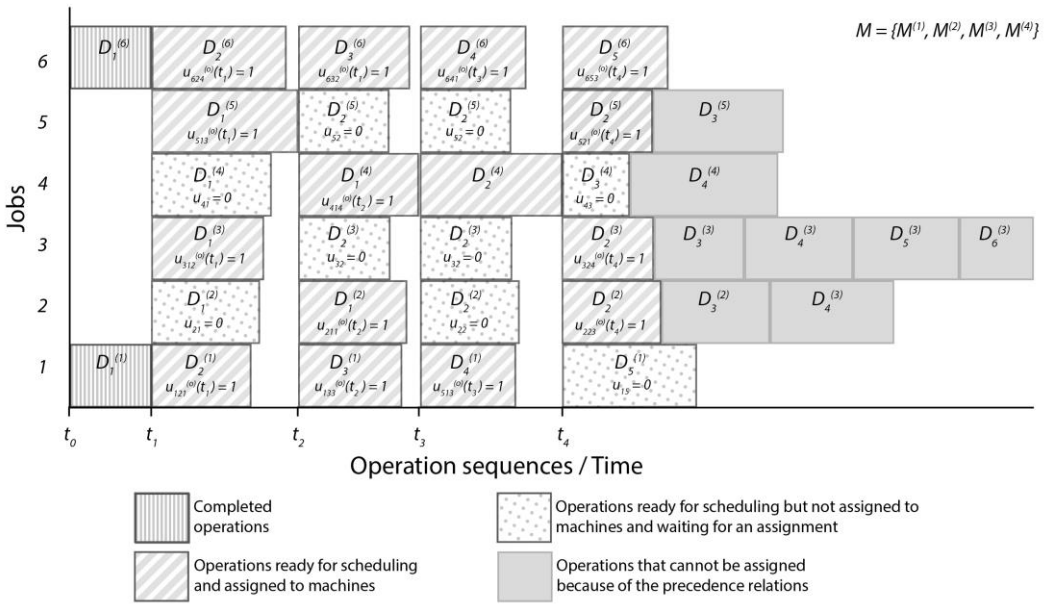


Fig. 4. Dynamic decomposition principle

To summarize and generalize the example in Fig. 4, the small dimensionality at each time point results from the dynamic decomposition described by the constraints on the precedence relations, i.e., at each time point we consider only those operations that can be assigned to machines at that time point, excluding those operations that have already been completed and those that cannot start because the predecessors are not completed yet. By dynamically decomposing the assignment matrix in time, we represent the optimal control problem as a two-point boundary problem, then to treat polynomially solvable problems (by tendency) of small dimensionality at each time point, and finally to integrate the partial solutions through the maximum principle by integrating the main and adjoint equation systems.

It is commonly known that analytical methods for an optimal control have been proven for small-dimensional systems. In the control engineering practice, numerical methods have been applied. Recent surveys of those methods have been presented by Dolgui et al. (2019) and Sokolov et al. (2018). A methodical challenge in applying the maximum principle is to find the coefficients of the adjoint system which change in dynamics. Another methodical challenge of boundary problems is that the initial conditions for the adjoint variables $\psi(T_0)$ are not given. At the same time, an optimal control should be calculated subject to the end conditions.

According to Dolgui et al. (2019), numerical methods can be classified into state space (i.e., direct methods, e.g., gradient methods), control space (i.e., indirect methods based on control variations such as the method of successive approximations), and trajectory space (e.g., dynamic programming method) methods. The system space method is based on the optimal control problem presentation as a two-point boundary problem using the maximum principle and is rooted in deriving necessary conditions of optimal control. Two systems of differential equations are solved: the main system and the conjugate one. This provides an optimal program control vector $\mathbf{u}^*(t)$ and the state trajectory $\mathbf{x}^*(t)$. The vector $\mathbf{u}^*(t)$ at time $t = T_0$ under the conditions $\mathbf{h}_0(\mathbf{x}(T_0)) \le \mathbf{O}$ and for the given value of $\boldsymbol{\psi}(T_0)$ returns the maximum to the Hamiltonian function. The missing initial conditions of the adjoint system are computed based on different algorithms, e.g., considering a constructive heuristic solution.

Considering the constraints (4)-(20), we suggest a combination of system and control space methods while integrating optimal control and mathematical programming. Optimal control is not used for solving the combinatorial problem, but rather for enhancing the existing mathematical programming algorithms regarding non-stationarity, flow control, and continuous material flows. Since the control variables are presented as binary variables, methods of discrete optimization are applied to combinatorial tasks within certain time intervals. The proposed specialized computational method is valid for special control system classes such as linear systems, where methods and algorithms for quadratic linear problems are applied, or when the optimal control problem is presented as discrete optimization.

4.2. Computational Procedure

The following algorithm has been developed and tested to compute an optimal solution. We refer to the study (Ivanov et al. 2016a) for an optimality and complexity analysis of the computational procedure that has been used in this study as an anchor. The computational procedure for the developed model (Eqs. (1)-(3)) is based on the integration of the main and adjoint equation systems subject to the maximization of the Hamiltonian (29)-(35):

$$H(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\psi}(t)) = H_1 + H_2 + H_3 + H_4 + H_5 + H_6 + H_7, \text{ where}$$

$$H_1 = \sum_i \sum_j \sum_\mu [\psi_{i\kappa}^{(\mathrm{O})} \varepsilon_{i\kappa j}(t) \Theta_{i\kappa j} + \psi_j^{(\mathrm{P})} + \varphi_{i\kappa j}^{(1)}] \cdot u_{i\kappa j}^{(\mathrm{O})} \tag{29}$$

$$H_2 = \sum_i \sum_\mu \sum_j [\psi_{i\kappa}^{(\Pi)}] \cdot u_{i\kappa j}^{(\Pi)} \tag{30}$$

$$H_3 = \sum_i \sum_\kappa \sum_j [\varphi_{i\kappa j}^{(1)} + \psi_{i\kappa}^{(\mathrm{O})} \cdot \varepsilon_{i\kappa j} \cdot \Theta_{i\kappa j}] \cdot \omega_{i\kappa j}^{(\mathrm{O})} \tag{31}$$

$$H_4 = \sum_i \sum_\kappa \sum_j \varphi_{i\kappa j}^{(2)} \cdot z_{i\kappa j} \tag{32}$$

$$H_5 = \sum_i \sum_\kappa \sum_j \varphi_{i\kappa j}^{(3)} \cdot \nu_{i\kappa j} \tag{33}$$

$$H_6 = \sum_i \sum_\kappa p_{i\kappa}^{(B,1)} \cdot u_{i\kappa}^{(B,1)} \tag{34}$$

$$H_7 = \sum_i \sum_\mu p_{i\kappa}^{(B,2)} \cdot u_{i\kappa}^{(B,2)} \tag{35}$$

where $\boldsymbol{\psi}(t)$ is the adjoint vector.

The algorithm is formally introduced as follows.

Algorithm 1: DYN-CONTROL

0. Set $t = 0$, and initialize all parameters.
1. From the time point $t = t_0$ onwards, determine the control $\boldsymbol{u}^{(r+1)}(t)$ ($r = 0,1,2,\ldots$ denotes the number of the iteration).
2. For the given initial boundary conditions (21) and $\boldsymbol{\psi}(t_0)$, compute $\boldsymbol{u}^*(t)$ at t=t_0 to maximize (29)-(35).
   a) Solve the assignment problem: solve the maximization of the Hamiltonian H_1 for model (1), (3) with constraints (7), (20).
   b) Solve the linear programming problem: solve the maximization of the Hamiltonian H_2 for model (2) with constraints (15), (18), (19).
   c) Operation selection: select operations and constraints which meet the requirements (7), (18), (19), (20).
3. $\boldsymbol{u}^*(t_0)$ is then put into (1)-(6) and (36)-(53)
4. Main and adjoint systems are integrated.
5. The transversality conditions (54)-(61) are evaluated
6. Update $t \to t_1$.
7. $\boldsymbol{u}^*(t_1)$ is computed based on Hamiltonian (29)-(35) maximization, where $t_1 = t_0+\Delta$ (Δ is the chosen integration step). $\boldsymbol{x}^{(r)}(t_1)$ is generated.
8. If $t = t_f$, then the record value $J_G = J_G^{(r)}$ can be calculated.
9. The integration process is continued as long as the end boundary conditions (22) are reached and the convergence for J_G is realized.
10. The resulted $\boldsymbol{u}^{(*)}(t)$ is the optimized schedule.

Consider the algorithmic realization in more detail. The continuous optimization part can be described as follows. In order to compute the optimal program control vector $\boldsymbol{u}^*(t)$ and the corresponding state trajectory $\boldsymbol{x}^*(t)$, the main (1)-(6) and adjoint (36)-(52) systems need to be solved. In accordance with the maximum principle, the adjoint system $= \left\| \psi_{i\kappa}^{(\mathrm{O})} \psi_{i\kappa}^{(\Pi)} \psi_j^{(\mathrm{P})} \varphi_{i\kappa j}^{(1)} \varphi_{i\kappa j}^{(2)} \varphi_{i\kappa j}^{(3)} p_{i\kappa}^{(B,1)} p_{i\kappa}^{(B,2)} \right\|^T$ can be written corresponding to the main system (1)-(6) $\boldsymbol{x} = \left\| x_{i\kappa}^{(\mathrm{O})} x_{i\kappa}^{(\Pi)} x_j^{(\mathrm{P})} z_{i\kappa j} h_{i\kappa j} g_{i\kappa j} x_{i\kappa}^{(B,1)} x_{i\kappa}^{(B,2)} \right\|^T$ (see Appendix 2). The transversality conditions can be formulated in the way shown in Appendix 3. The analytical form of the computation of the coefficients is presented in Ivanov et al. (2016a).

For the given initial boundary conditions (21) and $\boldsymbol{\psi}(t_0)$, the $\boldsymbol{u}^*(t)$ at t=t_0 is computed to maximize (29)-(35). $\boldsymbol{u}^*(t_0)$ is then put into (1)-(6) and (36)-(52) and the main and adjoint systems are integrated.

The optimal program control vector at time $t = t_0$ for the given value of $\boldsymbol{\psi}(t)$ returns the maximum to (29)-(35), which can be transformed into a general performance index and expressed in scalar form J_G. Next, $\boldsymbol{u}^*(t_1)$ is computed based on Hamiltonian (29)-(35) maximization, where $t_1 = t_0+\Delta$ (Δ is the chosen integration step). As a result of the dynamic model run at each iteration, $\boldsymbol{x}^{(r)}(t)$ is generated. The adjoint system (36)-(53) (see Appendix 2) is integrated subject to $\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t)$ and over the interval from $t = t_f$ to $t = t_0$. In addition, if $t = t_f$, then the record value $J_G = J_G^{(r)}$ can be calculated. After this, the transversality conditions (54)-(61) are evaluated (see Appendix 3). The dynamic coordination parameters are the adjoint variables which change their values during the iterative process of the solution of the corresponding two-point boundary problem. The integration process is continued as long as the end boundary conditions (22) are reached and the convergence for J_G is reached.

Now let us turn to the discrete optimization part. Since the stations are flexible and can process multiple operations, the assignment problem exists. The maximization of the Hamiltonian H_1 (Eq. 29) for model (1), (3) in combination with the constraints (7), (20) solves the assignment problem. The maximum number of operations that can be assigned to a station is determined in Eq. (7). Due to the dynamic decomposition principle state above, the assignment matrix at each time point is smaller.

The maximization of the Hamiltonian H_2 (Eq. 30) for model (2) in combination with the constraints (15) (18), (19) solves the linear programming problem. At each time point, only operations and constraints which meet the requirements (7), (18), (19), (20) are considered. By a dynamic switching of constraints (8)-(17), the size of the scheduling problem at each time point is reduced. The Hamiltonians (30) and (31) can be maximized when the constraints (13) and (19) satisfy the corresponding variables $\omega_{i\kappa j}^{(\mathrm{O})}$ and $u_{i\kappa j}^{(\Pi)}$. In this case, only a part of the constraints (8)-(17) is considered for the current assignment problem since, when the control in (8)-(17) is switched to 1, it becomes active in the right-hand part of the equations (1), (2), (3). Therefore, the reduction of the problem dimensionality at each time point in the calculation process is ensured because of the recurrent operation description. At each time point, the Hamiltonian function is extremized. The locally coordinated sub-problems are partial combinatorial assignment problems and linear programming problems which are formed dynamically depending on the precedence relations, the channel availability at the stations, and the operation execution progress.

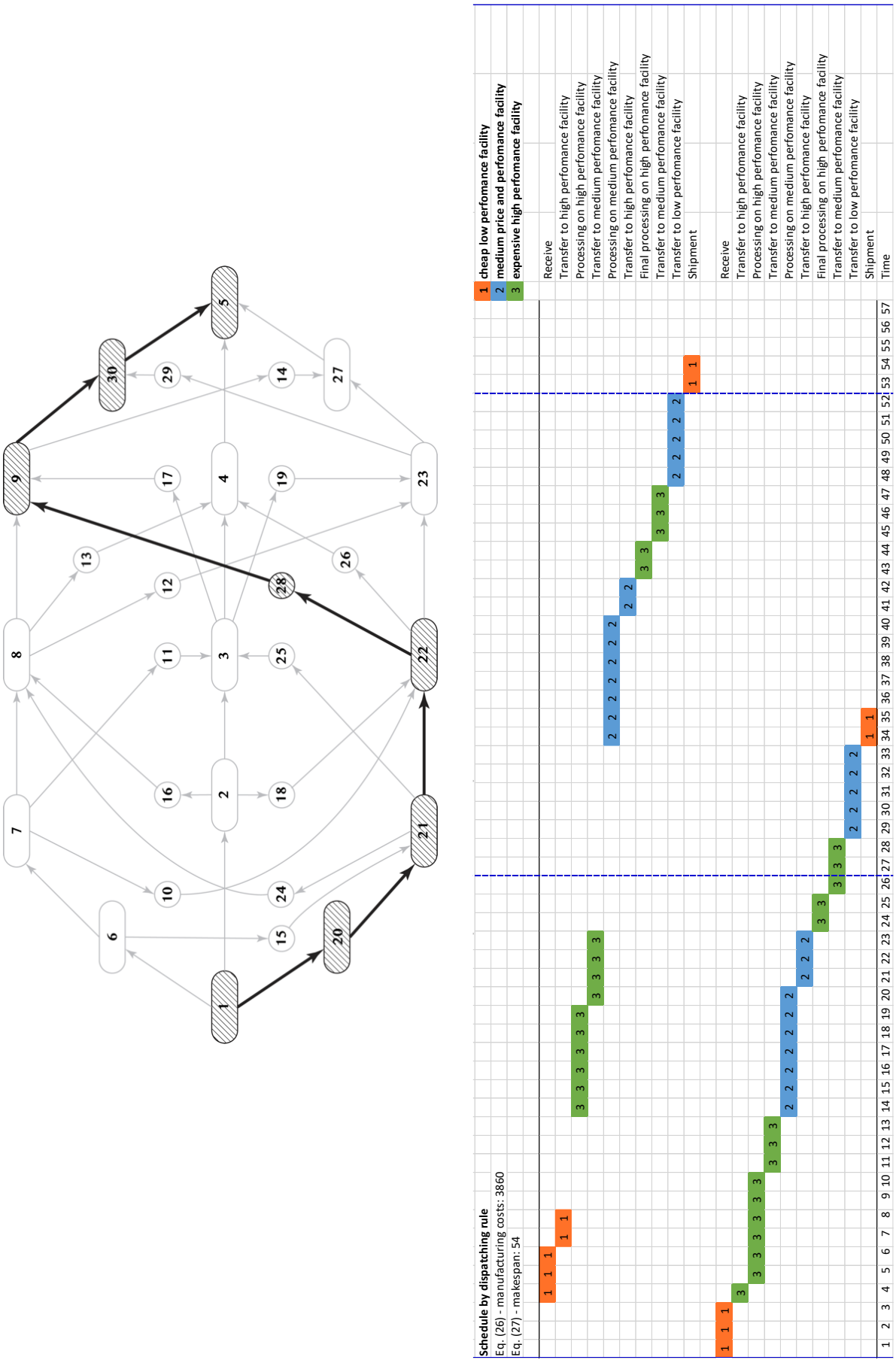
In the algorithm, the search for an optimal control in each iteration is performed in the class of boundary (e.g., pointwise or relay) functions which correspond to the discrete nature of decision making in scheduling. The formulated model is a linear, non-stationary, finite–dimensional controlled system 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. The algorithm optimality and sufficiency proofs can be found in the study (Ivanov et al. 2016a).

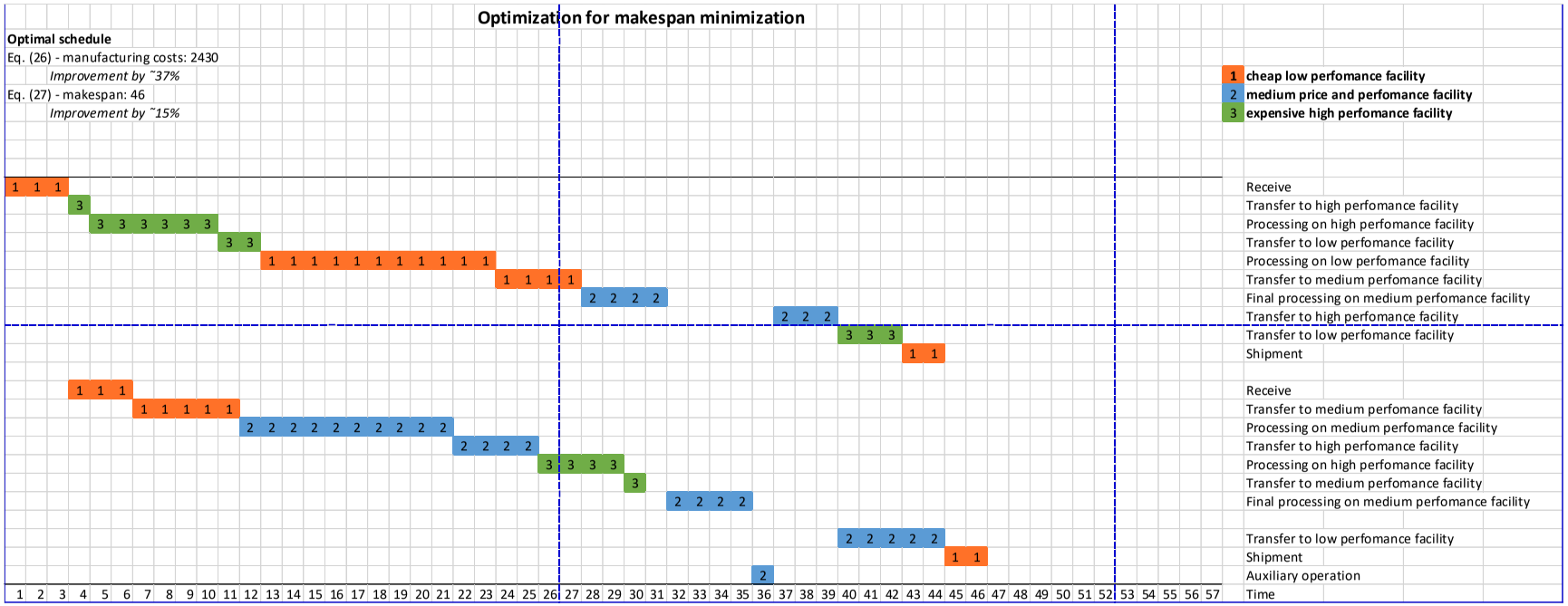
4.3. Illustrative Example

Consider an example with two jobs, three stations of different performance (slow and cheap, medium, fast and expensive), and three operations of each job. The processing of the operations at the station differs regarding the costs and time which results in a trade-off of manufacturing costs (Eq. 26) and makespan (Eq. 27). In Figs 5 and 6, a real-life example of a digital manufacturing system in the ship yard industry modelled using the developed control approach is depicted as a Gantt chart for two jobs. Nodes 1 and 5 are the source and sink nodes of the control trajectories. Nodes 7, 8, and 9 correspond to the operation execution at the medium performance station 2 (M_2). Nodes 2, 3, and 4 correspond to the operation execution at the slow and cheap station 1 (M_1). Nodes 21, 22, and 23 correspond to operation execution at the fast and expensive station 3 (M_3).

The algorithmic procedure is launched by an arbitrary feasible schedule $\tilde{\boldsymbol{u}}(t), t \in (T_0, T_f]$ according to the initial conditions (21) (Fig. 5). Fig. 6 displays the optimal control trajectories obtained for the example from Fig. 5 for an optimization against costs and makespan. Two jobs take a different control depending on the preferences in the objective function in terms of costs vs makespan minimization. In the case of makespan minimization (Fig. 6a), we have manufacturing costs (Eq. 26) of \$2430 in the optimal schedule which represents an improvement in comparison with the initial schedule generated by a dispatching rule at Step 1 of the algorithm by about 37%. The makespan (Eq. 27) is 46 time units, i.e., it is improved by about 15% as compared to the initial schedule generated. In the case of cost minimization (Fig. 6b), the manufacturing costs (Eq. 26) of \$1480 can be observed resulting in an improvement in comparison with the initial schedule by about 62%. The makespan is 67 time units implying a longer

time-to-customer by about 20% as compared to the initial schedule generated. The cheap and slow station M_1 is mostly used in the case of cost minimization whereas a mix of different stations can be observed in the schedule computed for the makespan minimization.





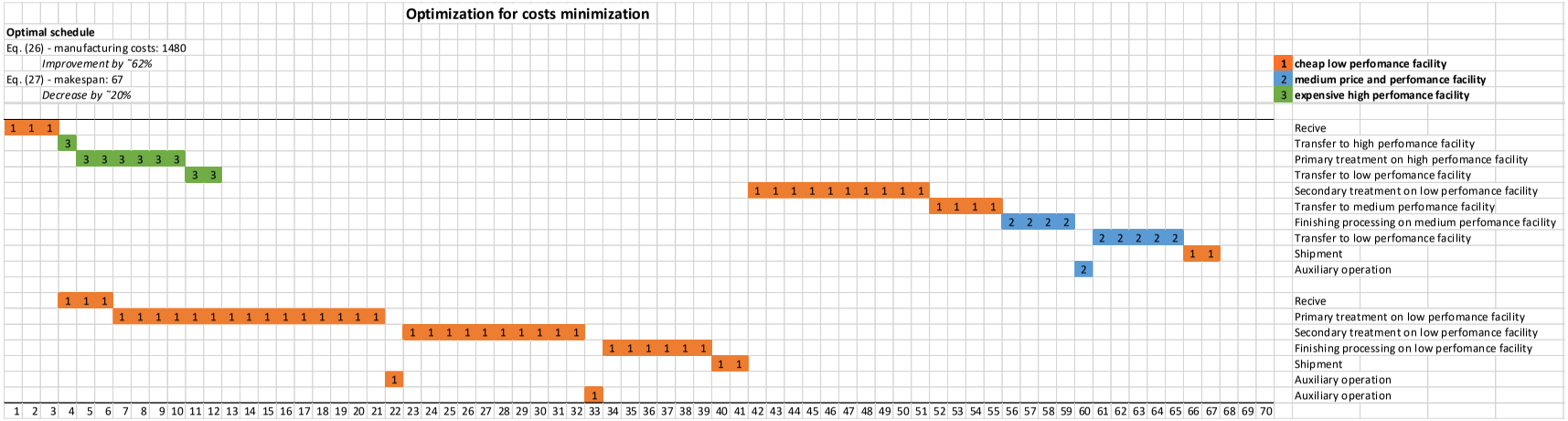


Fig. 5. Initial schedule obtained with a dispatching rule

a) Makespan minimization b) Cost minimization

Fig. 6. Optimal schedule obtained by control algorithm

5. Conclusions

The new technologies enable manufacturing with highly customized assembly systems and create a number of new remarkable challenges and opportunities for scheduling. In particular, the manufacturing processes for different customer orders may have individual machine structures such that the flexible stations are able to execute different functions subject to individual sets of operations within the jobs. Therefore, the problem of a simultaneous structural-functional synthesis of the systems with task and service compositions arises that in turn, feeds the challenging scheduling problems with dynamic hybrid structural-logical-terminal constraints. Our investigation accounts for this kind of scheduling problems with mixed structural-terminal-logical constraints when product and process are created at the same time. Examples of such systems have been increasingly encountered in digital manufacturing, e.g., in cloud manufacturing and Industry 4.0.

Our approach allows for revealing several insights which are difficult to receive with the help of other methods. First, the dynamic control model and dynamically changing constraints with the use of an original decomposition principle allow implicitly observing and controlling the variety of the process designs and schedule alternatives. Second, the constructed dynamic decomposition principle allows for dynamic control of constraints which appear and disappear in time from the dimensionality of the discrete optimization problems in the course of computational procedure due to precedence relations. Third, the proposed modelling paradigm makes it possible to consider the real-time information in the control framework as it becomes available with the help of updating the continuous state variables. In summary and in contrast to previous studies which assumed a fixed process design, our approach is capable of designing simultaneously the manufacturing process in regard to the available alternative stations, their current capacity utilization and the processing time, and the sequencing of the jobs. The use of state control variables in our model allows for updates in operations status that in turn, feeds automated information feedbacks and control of schedule execution and disruption detection. The latter can be used

for launching the re-scheduling procedure, comprehensively combining the planning and adaptation decision within a unified methodological framework of dynamic control theory. In addition, a wide range of analysis tools from control theory regarding stability, controllability, adaptability, etc. may be used for the schedule analysis when described in terms of control.

In the future, a robustness analysis of the overall system, i.e., both of the process design and the schedule, can extend the results of this study. In addition, computational examples may help to reveal new insights. A more detailed analysis of the Industry 4.0 technology may illuminate a taxonomy of structural-functional problems in this emerging research field. Finally, Industry 4.0 and digital technologies open new possibilities to the implementation of dynamic scheduling techniques using real-time data about the machine utilization, entering of new jobs and the operation processing status. This makes it possible to extend the algorithmic dynamic decomposition principles of the control scheduling models, e.g., by incorporating an event-oriented decomposition based on such events as “a machine becomes available”, “an operation is completed” or “a new job enters the system”. These extensions can be considered in light of future research topics.

References

Ahn, G., Park Y.-J., & S. Hur (2018) Performance computation methods for composition of tasks with multiple patterns in cloud manufacturing, International Journal of Production Research, DOI: 10.1080/00207543.2018.1451664

Aicardi M., Giglio D., Minciardi R. (2008). Optimal strategies for multiclass job scheduling on a single machine with controllable processing times. IEEE Trans on Automatic Control 53(2), 479-495.

Balakrishnan, A., J Geunes (2003). Production planning with flexible product specifications: an application to specialty steel manufacturing. Operations Research 51 (1), 94-112.

Barnes, J.W., Chambers, J.B. (1995) Solving the job shop scheduling problem with tabu search. IIE Transactions, 27(2): 257-263.

Battaïa, O., A Otto, F Sgarbossa, E Pesch (2018). Future trends in management and operation of assembly systems: from customized assembly systems to cyber-physical systems. Omega 78, 1-4.

Battaïa, O., Dolgui, A., Guschinsky, N. (2017) Decision support for design of reconfigurable rotary machining systems for family part production. Int Journal of Production Research, 55(5), 1368-1385.

Blazewicz J., Pesch E., Trystram D., Zhang G. (2015) New Perspectives in Scheduling Theory, Journal of Scheduling, 18(4), 333 – 334.

Blazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Sterna, M., Weglarz, J. (2019). Handbook on Scheduling. Springer.

Bortolini M., Galizia F. G., Mora C. (2018) Reconfigurable manufacturing systems: Literature review and research trend. J Manuf Syst 49, 93–106.

Bożek, A., Werner, F. (2018) Flexible job shop scheduling with lot streaming and sublot size optimisation. International Journal of Production Research, 6391-6411

Burgess K.; K.M. Passino (1999). Path-clearing policies for flexible manufacturing systems. IEEE Transactions on Automatic Control, 44(3), 573-578.

Chen, W., Song, J., Shi, L., Pi, L., & Sun, P. (2013) Data mining-based dispatching system for solving the local pickup and delivery problem, Annals of Operations Research, 203(1), 351–370.

Chen, W., Wang, X., Petersen, J., Tyagi, R., & Black, J. (2013) Optimal scheduling of demand response events for electric utilities, IEEE Transactions on Smart Grid, 23(4), 2309–2319.

Delorme X., Dolgui, A., Kovalyov, M.Y. (2012) Combinatorial design of a minimum cost transfer line, Omega - International Journal of Management Science, 40(1), 31–41.

Dolgui, A., Proth J.M. (2010). Supply Chain Engineering: Useful Methods and Techniques. Springer.

Dolgui, A., Ivanov, D., Sethi S.P., Sokolov, B. (2019) Scheduling in production, supply chain and Industry 4.0 systems by optimal control. International Journal of Production Research, 57(2), 411-432.

Feng, Y. Y., H. Yan. 2000. Optimal production control in a discrete manufacturing system with unreliable machines and random demands. IEEE Transactions on Automatic Control, 35, 2280-2296.

Frazzon, E.M., Kück, M., Freitag, M. (2018). Data-driven production control for complex and dynamic manufacturing systems. CIRP Annals, https://doi.org/10.1016/j.cirp.2018.04.033

Han, D., Wu J., Zhang H., Shi L. (2017). Optimal sensor scheduling for multiple linear dynamical systems. Automatica, 75, 260-270.

Ivanov D, Sokolov B, Kaeschel J (2010) A multi-structural framework for adaptive supply chain planning and operations with structure dynamics considerations. European Journal of Operational Research, 200, 409-420.

Ivanov D., Dolgui A., Sokolov B. (2016b) Robust dynamic schedule coordination control in the supply chain. Computers and Industrial Engineering, 94, 18-31.

Ivanov, D., Sethi S., Dolgui A., Sokolov, B. (2018). A survey on the control theory applications to operational systems, supply chain management and Industry 4.0. Ann Rev in Control, 46, 134-147.

Ivanov, D., Sokolov, B. (2010), Adaptive Supply Chain Management, Springer, London et al.

Ivanov, D., Sokolov, B. (2012). Dynamic supply chain scheduling, J of Scheduling 15(2), 201-216.

Ivanov, D., Sokolov, B., Dolgui, A., Werner, F., Ivanova, M. (2016a) A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory Industry 4.0. International Journal of Production Research, 54(2), 386-402.

Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. (2009). A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Computers & Operations Research, 36(2), 358–378.

Kaskavelis, C.A., Caramanis, M.C. (1998) Efficient Lagrangian relaxation algorithms for industry size job-shop scheduling problems. IIE Transactions, 30(11): 1085-1097.

Koren Y, Wang W, Gu X. (2017) Value creation through design for scalability of reconfigurable manufacturing systems. Int J Prod Res 55(5), 1227-1242

Kreter S., Rieck J., Zimmermann J. (2016). Models and solution procedures for the resource-constrained project scheduling problem with general terminal constraints and calendars. European Journal of Operational Research, 251 (2), 387-403

Kusiak, A. (2018) Smart Manufacturing, Int Journal of Production Research. 56(1-2), 508–517.

Kyparisis, G.J., & Koulamas, C.P. (2006) Flexible flow shop scheduling with uniform parallel machines. European Journal of Operational Research, 168, 985–997.

Li, J. SM Meerkov (2009). Production systems engineering. Springer Science & Business Media.

Liu, Y., Wang, L., Wang, X.V., Xu X. & L. Zhang (2018) Scheduling in cloud manufacturing: state-of-the-art and research challenges, International Journal of Production Research, DOI: 10.1080/00207543.2018.1449978

Lobo, B.J., Wilson, J.R., Thoney, K.A., Hodgson, T.J., King, R.E. (2014) A practical method for evaluating worker allocations in large-scale dual resource constrained job shops. IIE Transactions, 46(11): 1209-1226.

Mati Y., Xie X. (2011). Multiresource shop scheduling with resource flexibility and blocking. IEEE Transactions on Automation Science and Engineering, 8(1), 175-189.

Nayak, A., Reyes Levalle, R., Lee, S., & Nof, S.Y. (2016) Resource sharing in cyber-physical systems: modelling framework and case studies. Int Journal of Production Research, 54(23), 6969-6983.

Nielsen, I., Q.-V. Dang, G. Bocewicz, and Z. Banaszak (2017). A Methodology for Implementation of Mobile Robot in Adaptive Manufacturing Environments. Journal of Intelligent Manufacturing 28(5): 1171–1188.

Park, SJ, Yang JM. (2009), Supervisory control for real-time scheduling of periodic and sporadic tasks with resource constraints. Automatica, 45(11), 2597-2604

Pinedo M. (2018). Scheduling: Theory, Algorithms, and Systems. Springer, New York, 5th Ed.

Rossit, D.A., Tohmé F., & M. Frutos (2018) Industry 4.0: Smart Scheduling, International Journal of Production Research, DOI: 10.1080/00207543.2018.1504248

Sethi SP., G. Sorger ; XY Zhou (2000) Stability of real-time lot-scheduling and machine replacement policies with quality levels. IEEE Transactions on Automatic Control 45(11), 2193-2196.

Shen, L., Dauzere-Peres, S., Neufeld, J. (2018) Solving the flexible job shop scheduling problem with sequence-dependent setup times. European Journal of Operational Research, 265, 503 – 516.

Siemens (2018). https://www.siemens.com, accessed on September 16, 2018.

Sokolov B., Dolgui, A., Ivanov, D. (2018). Optimal control algorithms and their analysis for short-term scheduling in manufacturing systems. Algorithms, 11(5), 57.

Tahar, D.N., Yalaoui, F., Chu, C., & Amodeo, L. (2006) A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times. International Journal of Production Economics, 99(1–2), 63–73.

Tao S., Dong S. (2017). Scheduling resource-constrained project problem with alternative activity chains. Computers & Industrial Engineering, 114, 288-296.

Tao, F., Qi, Q., Liu, A., Kusiak, A. (2018). Data-driven smart manufacturing. Journal of Manufacturing Systems, https://doi.org/10.1016/j.jmsy.2018.01.006

Tran T.H, Ng K.M. (2011). A water-flow algorithm for flexible flow shop scheduling with intermediate buffers. J Sched (2011) 14:483–500

Vallada, E., Ruiz, R. & Framinan, J. M. (2017). New hard benchmark for flowshop scheduling problems minimising makespan. European Journal of Operational Research, 240, 666-677.

Vincent LVH; S. G. Ponnambalam (2013). A differential evolution-based algorithm to schedule flexible assembly lines. IEEE Transactions on Automation Science and Engineering 10(4), 1161-1165.

Xu, J., Tran, H.M., Gautam, N., Bukkapatnam, S.T.S. (2019) Joint production and maintenance operations in smart custom-manufacturing systems. IISE Transactions, forthcoming.

Yang, H., Kumara, S., Bukkapatnam, S.T.S, Tsung, F. (2019) The internet of things for smart manufacturing: A review. IISE Transactions, forthcoming.

Youssef AM, ElMaraghy HA. (2006) Modelling and optimization of multiple-aspect RMS configurations. Int J Prod Res 46(22), 4929–4458.

Yu, W., Liu, Z., Wang, L., & Fan, T. (2011) Routing open shop and flow shop scheduling problems. European Journal of Operational Research, 213(1), 24–36.

Zhong, R.Y., Xu, C., Chen, C., Huang, G.Q. (2017) Big data analytics for physical Internet-based intelligent manufacturing shop floors. International Journal of Production Research, 55(9), 2610-2621.

Appendix 1 Notations

Table A1 Sets, maps, constants

Notation	**Meaning**
$A = \{A_i, i = 1, \dots, n\}$	Set of customer orders (jobs)
$D^{(i)} = \left\{D_\kappa^{(i)}, \kappa = 1, \dots, S_i\right\}$	Set of operations
$M = \left\{M_j, j = 1, \dots, m\right\}$	Set of stations
$P = \left\{P_\kappa^{(i)}, \kappa = 1, \dots, S_i\right\}$	Set of material, energy or information flows in the manufacturing process.
U	Set of feasible control inputs
J	Set of performance indicators
$\Pi^t_{<\delta,\delta'>}$	Map describing the allowable transitions from one multi-structural macro-state to another one
Δ	Set of dynamic and static alternatives of the manufacturing process
$\mathbf{X}(\boldsymbol{\xi}(t), t)$	Area of the allowable states of the assembly line structural dynamics

Table A2 Parameters and functions

Notation	**Meaning**
$a_{i\kappa}^{(\mathrm{O})}$, $a_{i\kappa}^{(\Pi)}$	Planned manufacturing and transportation quantities for each operation.
t_0	Start time of the scheduling horizon
t_f	End time of the scheduling horizon
$c_{i\kappa j}^{(\Pi)}$	Maximum processing rate for the operation $D_\kappa^{(i)}$ at the M_j
μ_j	Manufacturing costs per time unit at the M_j
$\xi(t)$	Vector of perturbation impacts

$\varepsilon(t)$	Preset matrix time function of the time-spatial constraints for the stations
$\Theta_{i\text{æ}j\lambda}(t)$	Preset matrix time function of the time-spatial constraints for the channels

Table A3 Indices

Notation	**Meaning**
i	Job index
n	Running numbers of a job
k	Operation index
s	Running numbers of an operation
j	Station index
m	Running numbers of a station
r	Number of the iteration of the algorithm
α,β	Indices of the precedence relations “and”/“or”
(O), (Π), (P),	Indices to describe the relationships of the respective sets to the operations, flows, and machines
t	Current time point

Table A4 Decision control and state variables

Notation	**Meaning**
$\mathbf{u}(t)$	Optimal control vector characterizing the production schedule
$u_{i\kappa j}^{(\Pi)}$	Control variable characterizing the production rate
$x_{i\kappa}^{(O)}$	State variable characterizing the lead time for operation a $D_{\kappa}^{(i)}$ at each moment t

$x_{i\kappa}^{(\Pi)}$	State variable characterizing the flow time of the operation $D_{\kappa}^{(i)}$
$x_j^{(\mathrm{P})}$	State variable characterizing the run time of the station M_j
$v, \omega^{(\mathrm{O})}, u^{(B,1)}, u^{(B,2)}, z, h, g, x^{(B,1)}, x^{(B,2)}$	auxiliary variables
$\sigma_{i\overline{\alpha}}^{(and)};\ \sigma_{i\overline{\beta}}^{(or)};\ \sigma_{i\overline{\overline{\beta}}}^{(or)};\ \sigma_{i\overline{\overline{\alpha}}}^{(and)};\ \sigma_{i\mu}^{(5,2)};$ $\sigma_{i\mu}^{(2,1)}$	Coefficients of the adjoint system
$\psi(t)$	Adjoint variable

Appendix 2 Adjoint system

$$\dot{\psi}_{i\kappa}^{(\mathrm{O})} = -\sum_{\overline{\alpha}\in\Gamma_{i\kappa}^{-}} \sigma_{i\overline{\alpha}}^{(1)} \cdot \sum_{j=1}^{m} u_{i\overline{\alpha}j}^{(\mathrm{O})} - \sum_{\overline{\beta}\in\Gamma_{i\kappa}^{-}} \sigma_{i\overline{\beta}}^{(2)} \cdot \sum_{j=1}^{m} u_{i\overline{\beta}j}^{(\mathrm{O})} \cdot \prod_{\substack{\delta\in\Gamma_{i\overline{\beta}}^{+} \\ \delta\neq\kappa}} x_{i\delta}^{(\mathrm{O})} \tag{36}$$

$$\dot{\psi}_{i\kappa}^{(\Pi)} = -\sum_{\overline{\alpha}\in\Gamma_{i\kappa}^{+}} \sigma_{i\overline{\alpha}}^{(1)} \cdot \sum_{j=1}^{m} u_{i\overline{\alpha}j}^{(\mathrm{O})} - \sum_{\overline{\beta}\in\Gamma_{i\kappa}^{+}} \sigma_{i\overline{\beta}}^{(2)} \cdot \sum_{j=1}^{m} u_{i\overline{\beta}j}^{(\mathrm{O})} \cdot \prod_{\substack{\mu\in\Gamma_{i\overline{\beta}}^{-} \\ \mu\neq\kappa}} \left(a_{i\mu}^{(\Pi)} - x_{i\mu}^{(\Pi)}\right) - \tag{37}$$

$$-\sigma_{i\kappa}^{(2,1)} \cdot \textstyle\sum_{j=1}^{m} \omega_{i\kappa j}^{(\mathrm{O})} - \sigma_{i\kappa}^{(5,2)} \cdot u_{i\kappa}^{(B,2)}$$

$$\dot{\psi}_{j}^{(\mathrm{P})} = 0 \tag{38}$$

$$\dot{\varphi}_{i\kappa j}^{(1)} = -\varphi_{i\kappa j}^{(2)} - \sigma_{i\kappa j}^{(4,1)} \cdot \nu_{i\kappa j} \tag{39}$$

$$\dot{\varphi}_{i\kappa j}^{(2)} = 0 \tag{40}$$

$$\dot{\varphi}_{i\kappa j}^{(3)} = -\sigma_{i\kappa}^{(2,1)} \cdot \omega_{i\kappa j}^{(\mathrm{O})} \tag{41}$$

$$\dot{p}_{i\kappa}^{(B,1)} = 0;\ \dot{p}_{i\kappa}^{(B,2)} = 0 \tag{42}$$

$$\sigma_{i\overline{\alpha}}^{(1,1)} = \frac{\left(\psi_{i\overline{\alpha}}^{(\mathrm{O})} \varepsilon_{i\overline{\alpha}j}(t)\Theta_{i\overline{\alpha}j} + \psi_{j}^{(\mathrm{P})} + \varphi_{i\overline{\alpha}j}^{(1)}\right) + \sigma_{i\overline{\alpha}j}^{(3,1)} c_{i\overline{\alpha}j}^{(\Pi)}}{1 - \sum_{\kappa\in\Gamma_{i\overline{\alpha}}^{+}} x_{i\kappa}^{(\mathrm{O})}} \tag{43}$$

$$\sigma_{i\overline{\beta}}^{(1,2)} = \frac{\left(\psi_{i\overline{\beta}}^{(\mathrm{O})} \varepsilon_{i\overline{\beta}j}(t)\Theta_{i\overline{\beta}j} + \psi_{j}^{(\mathrm{P})} + \varphi_{i\overline{\beta}j}^{(1)}\right) + \sigma_{i\overline{\beta}j}^{(3,1)} c_{i\overline{\beta}j}^{(\Pi)}}{1 - \prod_{\kappa\in\Gamma_{i\overline{\beta}}^{+}} x_{i\kappa}^{(\mathrm{O})}} \tag{44}$$

$$\sigma_{i\overline{\alpha}}^{(1,3)} = \frac{\left(\psi_{i\overline{\alpha}}^{(\mathrm{O})} \varepsilon_{i\overline{\alpha}j}(t)\Theta_{i\overline{\alpha}j} + \psi_{j}^{(\mathrm{P})} + \varphi_{i\overline{\alpha}j}^{(1)}\right) + \sigma_{i\overline{\alpha}j}^{(3,1)} c_{i\overline{\alpha}j}^{(\Pi)}}{1 + \sum_{\kappa\in\Gamma_{i\overline{\alpha}}^{-}} \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right)} \tag{45}$$

$$\sigma_{i\overline{\beta}}^{(1,4)} = \frac{\left(\psi_{i\overline{\beta}}^{(\mathrm{O})} \varepsilon_{i\overline{\beta}j}(t)\Theta_{i\overline{\beta}j} + \psi_{j}^{(\mathrm{P})} + \varphi_{i\overline{\beta}j}^{(1)}\right) + \sigma_{i\overline{\beta}j}^{(3,1)} c_{i\overline{\beta}j}^{(\Pi)}}{1 - \prod_{\kappa\in\Gamma_{i\overline{\beta}}^{-}} \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right)} \tag{46}$$

$$\sigma_{i\kappa}^{(2,1)} = \frac{\psi_{i\kappa}^{(\mathrm{O})} \cdot \varepsilon_{i\kappa j}(t)\Theta_{i\kappa j} + \varphi_{i\kappa j}^{(1)}}{\left(\left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right) - g_{i\kappa j} + 1\right)} \tag{47}$$

$$\sigma_{i\kappa j}^{(3,1)} = \psi_{i\kappa}^{(\Pi)}, \text{ if } \rho_{i\kappa}^{(3,1)} = 0, \rho_{i\kappa}^{(3,2)} < 0 \tag{48}$$

$$\sigma_{i\kappa j}^{(3,2)} = -\psi_{i\kappa}^{(\Pi)}, \text{ if } \rho_{i\kappa}^{(3,1)} < 0, \rho_{i\kappa}^{(3,2)} = 0 \tag{49}$$

$$\rho_{i\kappa}^{(3,1)} = u_{i\kappa j}^{(\Pi)} - c_{i\kappa j}^{(\Pi)} u_{i\kappa j}^{(\mathrm{O})} \leq 0$$
$$\rho_{i\kappa}^{(3,2)} = -u_{i\kappa j}^{(\Pi)} \leq 0 \tag{50}$$

$$\sigma_{i\kappa j}^{(4,1)} = \frac{\varphi_{i\kappa j}^{(3)}}{1 + \left(a_{i\kappa}^{(\mathrm{O})} - \sum_{j=1}^{m} z_{i\kappa j}\right)} \tag{51}$$

$$\sigma_{i\kappa}^{(5,1)} = \frac{p^{(B,1)}}{1 - x_{i\kappa}^{(B,2)}} \tag{52}$$

$$\sigma_{i\kappa}^{(5,2)} = \frac{p^{(B,2)}}{1 + \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right)} \tag{53}$$

Appendix 3 Transversality conditions

$$\psi_{i\kappa}^{(\mathrm{O})}(t_f) = \left(a_{i\kappa}^{(\mathrm{O})} - x_{i\kappa}^{(\mathrm{O})}\right)\Big|_{t=t_f} \tag{54}$$

$$\psi_{i\kappa}^{(\Pi)}(t_f) = \left(a_{i\kappa}^{(\Pi)} - x_{i\kappa}^{(\Pi)}\right)\Big|_{t=t_f} \tag{55}$$

$$\psi_{i\kappa}^{(\mathrm{P})}(t_f) = \left(\tilde{t}_f - x_j^{(\mathrm{P})}(t_f)\right) \tag{56}$$

$$\varphi_{i\kappa j}^{(1)}(t_f) = a_{i\kappa}^{(\mathrm{O})} - \sum_{j=1}^{m} z_{i\kappa j}\Bigg|_{t=t_f} - \left(z_{i\kappa j} g_{i\kappa j} + \frac{{a_{i\kappa}^{(\mathrm{O})}}^2}{2} - h_{i\kappa j}\right) {z_{i\kappa j}}^2 - \tag{57}$$

$$- \left(z_{i\kappa j} g_{i\kappa j} + \frac{{a_{i\kappa}^{(\mathrm{O})}}^2}{2} - h_{i\kappa j}\right)^2 z_{i\kappa j}$$

$$\varphi_{i\kappa j}^{(2)}(t_f) = \left(z_{i\kappa j} g_{i\kappa j} + \frac{{a_{i\kappa}^{(\mathrm{O})}}^2}{2} - h_{i\kappa j}\right) {z_{i\kappa j}}^2 \tag{58}$$

$$\varphi_{i\kappa j}^{(3)}(t_f) = -\left(z_{i\kappa j} g_{i\kappa j} + \frac{{a_{i\kappa}^{(\mathrm{O})}}^2}{2} - h_{i\kappa j}\right) {z_{i\kappa j}}^2 \tag{59}$$

$$p_{i\kappa}^{(B,1)}(t_f) = \frac{x_{iS}^{(B,2)}(t_f)}{t_f - x_{i\kappa}^{(B,1)}(t_f)} \tag{60}$$

$$p_{i\kappa}^{(B,2)}(t_f) = \frac{1}{t_f - x_{i\kappa}^{(\mathrm{O})}} \tag{61}$$

where $\sigma_{i\bar{\alpha}}^{(1,1)}, \sigma_{i\bar{\beta}}^{(1,2)}, \sigma_{i\bar{\bar{\alpha}}}^{(1,3)}, \sigma_{i\bar{\bar{\beta}}}^{(1,4)}, \sigma_{i\kappa}^{(2,1)}, \sigma_{i\kappa j}^{(3,1)}, \sigma_{i\kappa j}^{(3,2)}, \sigma_{i\kappa j}^{(4,1)}, \sigma_{i\kappa}^{(5,1)}, \sigma_{i\kappa}^{(5,2)}$ are the computed coefficients that are similar to the Lagrange multipliers in mathematical programming, but which change dynamically.