Computers and Chemical Engineering 106 (2017) 621–644

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Mixed-integer programming models for simultaneous batching and scheduling in multipurpose batch plants

Hojae Lee, Christos T. Maravelias*

Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, WI 53706, USA

ARTICLE INFO

Article history:
Received 10 March 2017
Received in revised form 29 June 2017
Accepted 22 July 2017
Available online 2 August 2017

Keywords:
Sequential production environment
Storage policies

ABSTRACT

We propose two novel discrete-time mixed-integer programming models for simultaneous batching and scheduling in multipurpose batch plants with storage constraints. The proposed models adopt two different modeling approaches. The first is based on explicit labeling of batches, while the second is based on identifying possible batch size intervals for each order and the corresponding unit routings. We also present extensions that allow us to consider limited shared utilities (with both fixed and time-varying availability and cost), storage with capacity limits and stage-dependent batch sizes. Finally, we study how instance characteristics (e.g. expected number of batches per order, uniformity in unit capacities) impact the effectiveness of the proposed models. We show that by carefully selecting the model allows us to effectively solve large-scale instances.

1. Introduction

Chemical manufacturing facilities consist of a series of operations that compete for limited resources (e.g. process units, utilities). Through production scheduling, favorable allocation of these limited resources to various production tasks over time can be achieved (Pinedo, 2012). Although production scheduling was originally regarded as a feasibility problem, optimization techniques allow for a more rigorous decision making that optimally utilizes available resources, while pursuing objectives that concern customer satisfaction, operation costs and profits. Consequently, production scheduling through exact mathematical programming methods has been the subject of active research in the Process Systems Engineering (PSE) community in the past two decades (Reklaitis, 1996; Pinto and Grossmann, 1998; Kallrath, 2002; Floudas and Lin, 2004; Mendez et al., 2006; Maravelias, 2012; Harjunkoski et al., 2014).

Among the various production environments studied, the sequential environment is one of the most common in batch manufacturing facilities. It exhibits a series of stages in which batches have to go through without being mixed, split or recycled. Among problems in sequential environments, multistage batch plants are the facilities that have received the most attention. However, only a limited amount of studies have been conducted on the more general multipurpose batch plants.

Furthermore, most of the existing approaches for scheduling in sequential environments are based on several restrictive assumptions. Specifically, batching decisions are assumed to be decoupled from scheduling decisions, often leading to suboptimal solutions. This assumption may be reasonable in relatively simple processes, where good batching decisions can be readily identified. However, as the complexity increases, making batching decisions without considering their scheduling aspects may lead to inefficient utilization of resources. Another simplifying assumption commonly made is that there are always storage vessels available for intermediate products, i.e. unlimited intermediate storage (UIS). However, this assumption is not applicable to many of the process industries.

The goal of this paper is to develop mathematical programming models for scheduling in multipurpose facilities that can address the aforementioned limitations. We develop two mixed-integer programming (MIP) models that adopt two different modeling approaches. Furthermore, we discuss the computational performance of the two approaches. This paper is structured as follows. In Section 2, we provide a literature review, focusing on sequential environment and the limitations of existing approaches. Next in Section 3, we formally define the problem we consider. Sections 4–6 provide the mathematical formulations of the proposed models, as well as extensions for limited shared utilities, storage constraints and stage-dependent batch sizes. In Section 7, several illustrative examples are presented

* Corresponding author.
E-mail address: christos.maravelias@wisc.edu (C.T. Maravelias).

http://dx.doi.org/10.1016/j.compchemeng.2017.07.007
0098-1354/

Indices/sets

Indices/sets

$i \in \mathbf{I}$	orders
$j \in \mathbf{J}/\mathbf{J}^P/\mathbf{J}^S$	units/processing units/storage vessels
$k \in \mathbf{K}$	stages
$l \in \mathbf{L}$	batches
$m \in \mathbf{M}$	batch size intervals
$r \in \mathbf{R}$	resources
$t \in \mathbf{T}$	time points/periods

Subsets

$\mathbf{I}_{jk}$	orders on stage k that can be processed in unit j
$\mathbf{I}^S_{jk}$	orders that can be stored in vessel j after being processed on stage k
$\mathbf{I}_{jkt}$	orders on stage k that can start in unit j at time point t
$\mathbf{J}_i$	process units that can process order i
$\mathbf{J}^S_i$	storage vessels that can store order i
$\mathbf{J}^{\max}_i/\mathbf{J}^{\min}_i$	remaining candidate units for batch size interval generation algorithm
$\mathbf{J}_{ik}$	process units that can process order i on stage k
$\mathbf{J}^S_{ik}$	storage vessels that can store a batch of order i that has been processed on stage k
$\mathbf{J}_{ikt}$	process units in which order i on stage k can start at time point t
$\mathbf{J}_{imk}$	process units that can process order i on stage k assigned to batch size interval m
$\mathbf{J}^S_{imk}$	storage vessels that can store order i on stage k assigned to batch size interval m
$\mathbf{K}_i$	stages on which order i is processed
$\mathbf{K}^-_{ik}/\mathbf{K}^+_{ik}$	stages on which order i have to be processed before/after stage k
$\mathbf{L}_i$	batches of order i
$\mathbf{M}_i$	batch size intervals for order i
$\mathbf{M}_{ij}$	batch size intervals for order i when processed in unit j
$\mathbf{M}_{ijk}/\mathbf{M}^S_{ijk}$	batch size intervals for order i on stage k processed/stored in unit/vessel j
$\mathbf{P}^{\max}_i/\mathbf{P}^{\min}_i$	remaining candidate units for algorithm with stage-dependent batch sizes
$\mathbf{T}_{ijk}$	set of feasible time points for order i on stage k to start in unit j
$\mathbf{T}^S_{ik}$	set of feasible time periods in which order i on stage k can be stored in any compatible vessel

Parameters

α_{ijk}	cost of processing a batch of order i on stage k in unit j
α^U_{rt}	cost of resource r during time period t
α^{Inv}_{ijk}	inventory cost of batch of order i on stage k stored in vessel j
$\beta^{\min}_j/\beta^{\max}_j$	minimum/maximum capacity of unit or vessel j
$\beta^{eff,\min}_i/\beta^{eff,\max}_i$	effective minimum/maximum batch sizes of order i
$\hat{\beta}^{eff,\min}_i$	realistic minimum batch size of order i
δ	time discretization length
ε_{ik}	earliest start time of order i on stage k
$\zeta^{\min}_{im}/\zeta^{\max}_{im}$	lower/upper endpoint of batch size interval m of order i
η^0/η^F	start/end of the scheduling horizon (i.e. $\eta^0 = min_{i\in\mathbf{I}}\mu_i$, $\eta^F = max_{i\in\mathbf{I}}\phi_i$)
Θ_{ij}	number of time ranges of order i on its last stage when processed in unit j (Eq. (37))
κ^{LS}_i	last stage of order i
λ_{ik}	latest finish time of order i on stage k
μ_i	release time of order i
ξ_i	amount due of order i
σ_{ik}	yield of order i on stage k
ρ_{ijkt}	utility r required to process a batch of order i on stage k in unit j
τ_{ijk}	processing time of order i on stage k in unit j
ϕ_i	due date of order i
χ_{ik}	relative size of the final product to batch size on stage k of order i
ψ_{rt}	availability of resource r during time period t

Binary variables

X^L_{iljkt}	=1 if batch $l \in \mathbf{L}_i$ on stage k starts in unit j at time point t
X^M_{imjkt}	=1 if order i on stage k assigned to interval m starts in unit j at time point t
X^S_{ijkt}	=1 if any batch of order i on stage k is being stored in vessel j during time period t