



A novel constraint programming model for large-scale scheduling problems in multiproduct multistage batch plants: Limited resources and campaign-based operation



Franco M. Novara^a, Juan M. Novas^b, Gabriela P. Henning^{c,*}

^a FIQ, Universidad Nacional del Litoral, Santiago del Estero 2829, Santa Fe 3000, Argentina

^b CIEM (Universidad Nacional de Córdoba, CONICET), Medina Allende s/n, Córdoba 5000, Argentina

^c INTEC (Universidad Nacional del Litoral, CONICET), Güemes 3450, Santa Fe 3000, Argentina

ARTICLE INFO

Article history:

Received 30 December 2015

Received in revised form 19 April 2016

Accepted 19 April 2016

Available online 20 May 2016

Keywords:

Scheduling

Constraint programming

Multiproduct multistage batch plants

Campaign operating mode

ABSTRACT

This contribution introduces an efficient constraint programming (CP) model that copes with large-scale scheduling problems in multiproduct multistage batch plants. It addresses several features found in industrial environments, such as topology constraints, forbidden product-equipment assignments, sequence-dependent changeover tasks, dissimilar parallel units at each stage, limiting renewable resources and multiple-batch orders, among other relevant plant characteristics. Moreover, the contribution deals with various inter-stage storage and operational policies. In addition, multiple-batch orders can be handled by defining a campaign operating mode, and lower and upper bounds on the number of batches per campaign can be fixed. The proposed model has been extensively tested by means of several case studies having various problem sizes and characteristics. The results have shown that the model can efficiently solve medium and large-scale problems with multiple constraining features. The approach has also rendered good quality solutions for problems that consider multiple-batch orders under a campaign-based operational policy.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Multiproduct multistage batch plants are characterized by the manufacturing of multiple products having similar recipes. These plants generally operate on the basis of a set of orders, each having its due date. In these environments, the short-term scheduling has an important impact on the costs and benefits of the organization, as well as on the effective use of the limited resources. Therefore, it is crucial to develop efficient scheduling approaches capable of solving big size problems and finding good quality solutions in reduced computational times.

In multiproduct multistage environments, each order is fulfilled by means of a single or multiple batches. Most of the academic contributions, such as (Harjunkoski and Grossmann, 2002; Gupta and Karimi, 2003; Marchetti and Cerdá, 2009a,b); among others, have assumed that a single batch per product is demanded. When multiple batches per order are required, it is usual to operate under a campaign mode. Thus, a given number of batches of the same product are sequentially processed in order to reduce changeover times and costs. The number and size of the batches required to satisfy a given order can be specified before the scheduling activity takes place by decoupling the batching and the scheduling decisions (sequential approach). On the contrary, the scheduling problem can include the definition of the number and size of batches (monolithic approach). However, the number of batches that comprise each product campaign is a decision that must be taken in any case.

This contribution introduces a novel constraint programming (CP) approach to address the short-term scheduling of multiproduct multistage batch plants which operate under a campaign mode. The approach focuses on unit assignment, batch sequencing and timing decisions, while optimizing a time-based or cost-based objective function. The proposal assumes that the batching decisions have been taken beforehand. It accounts for several features that are present in industrial environments, such as dissimilar parallel units at each stage, topology constraints, forbidden product-equipment assignments, sequence-dependent changeover tasks, intermediate due-dates,

* Corresponding author.

E-mail address: ghenning@intec.unl.edu.ar (G.P. Henning).

Nomenclature

Sets/indices

B/b	batches to be produced
B_p	batches of product p
C_p/c	set of all possible campaigns of product p . Its cardinality results from the expression $Card(B_p)/IB_p$
F_u	units belonging to stage $s + 1$, which are unconnected to unit u , belonging to stage s
F_p	set of product couples that correspond to forbidden production sequences $f = \langle p, p' \rangle$
$nbStages$	number of processing stages belonging to the production process
$P/p, p'$	demanded products
R/r	renewable resources (e.g. utilities, manpower, etc.)
S/s	processing stages
U/u	equipment units
U_s	units belonging to stage s

Parameters

$avail_r$	maximum availability of resource r
$avail_{CIP}$	maximum number of available cleaning-in-place devices
$changeOverTime$	changeover time associated with the triplet $\langle p, p', u \rangle$: the sequence dependent setup time between products p and p' on unit u
$cost_{p,u}$	processing cost of a batch of product p on unit u
dd_b	due-date of batch b
$fCost_u$	processing cost associated with the usage of unit u during the scheduling period
IB_p	minimum number of batches that a campaign of product p may comprise
$pt_{p,u}$	processing time required by a batch of product p on unit u
rd_u	ready time of unit u
$requir_{p,s,r}$	requirement of resource r in order to process a batch of product p at stage s
$cleanRequir_{b1,b2,u,r}$	requirement of resource r in order to perform the necessary cleaning between batches $b1$ and $b2$ in unit u
rt_b	release time of batch b
st_u	setup time of unit u
uB_p	maximum number of batches that a campaign of product p may comprise
$wt_{p,s}$	maximum waiting time for a batch of product p at stage s

Cumulative function

$UsageProfile_r$ accumulative usage of resource r as a function of time. It is employed to model the limited availability of resource r

Variables

$campaignTask_{c,p,u}$	campaign interval variable that spans over all the processing tasks that belong to a campaign c of product p carried out on unit u
$cleanTask_{b1,b2,u}$	optional interval variable that represents a cleaning activity between the processing tasks corresponding to batches $b1$ and $b2$ in unit u . When the campaign mode is considered, indexes $b1$ and $b2$ must be replaced by $c1$ and $c2$
$nTar_b$	binary variable that is equal to 1 when batch b finishes after its due-date, and 0 otherwise
$stageTask_{b,s}$	interval variable representing a processing task of batch b at stage s
tar_b	batch b tardiness
$task_{b,u,c}$	optional interval variable that represents a processing task of batch b in unit u , in the context of campaign c on unit u . If campaign mode is not considered, index c is ignored
$unitBatchSeq_u$	sequence variable defined for each unit u . It represents an ordering of task interval variables associated with u . Each interval variable (task) in this sequence is characterized by an attribute. When the changeovers between tasks are unit-independent, this attribute specifies the product p associated with each task variable. In case the changeovers are unit-dependent, the attribute represents both, the product and the unit where the task is assigned.
$unitCampaignSeq_u$	sequence variable defined for each unit u . It represents an ordering of campaign interval variables associated with u . Each interval variable (campaign) in this sequence is characterized by an attribute that specifies the product p associated with the campaign variable.
z_u	binary variable that is equal to 1 when unit u is used during the scheduling period, and 0 otherwise

different types of limiting renewable resources (e.g. utilities, manpower, cleaning-in-place devices, etc.), demanded by both processing and cleaning activities, multiple-batch orders (addressed in a campaign mode with consideration of a lower and upper bound on the number of batches per campaign). In addition, the approach is able to solve large-scale problems like the ones found in actual settings. The CP model is based on the ILOG-IBM OPL language and the CP Optimizer, which are embedded within the CPLEX Optimization Studio (IBM ILOG, 2013). It has been tested by means of a variety of problem instances having various sizes and characteristics.

The rest of the work is organized as follows: Section 2 presents the state-of-the-art. Section 3 describes the problem under study and Section 4 introduces the CP formulation. Finally, Section 5 discusses the computational results, and Section 6 presents the conclusions of this work.

Download English Version:

<https://daneshyari.com/en/article/171982>

Download Persian Version:

<https://daneshyari.com/article/171982>

[Daneshyari.com](https://daneshyari.com)