Chemical Engineering Science 136 (2015) 139–157

Contents lists available at ScienceDirect

Chemical Engineering Science

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

On the solution of large-scale mixed integer programming scheduling models

Sara Velez, Andres F. Merchan, Christos T. Maravelias *

Department of Chemical and Biological Engineering, University of Wisconsin—Madison, 1415 Engineering Dr., Madison, WI 53706, USA

HIGHLIGHTS

- We present extensions to discrete-time MIP model that allow to address realistic instances.
- We show how advanced solution methods can be extended and combined to address a wide range of problems.
- The proposed method yield (near) optimal solutions in time frames that can be used in practice.

ARTICLE INFO

Article history:
Received 3 November 2014
Received in revised form
4 April 2015
Accepted 12 May 2015
Available online 21 May 2015

Keywords:
Supply chain optimization
Solution methods
Reformulations
Tightening constraints
Process operations
Optimization

ABSTRACT

In this paper, we show how four recently developed modeling and solution methods can be integrated to address mixed integer programs for the scheduling of large-scale chemical production systems. The first method uses multiple discrete time grids. The second adds tightening constraints that lower bound the total production and number of batches for each task and material based on the customer demand, while the third generates upper bounding constraints based on inventory and resource availability. The final method is a reformulation that introduces a new integer variable representing the total number of batches of a task. We apply the aforementioned methods to large-scale problems with a variety of processing features, including variable conversion coefficients, changeovers, various storage policies, continuous processing tasks, setups, and utilities, using a discrete-time model. We illustrate how these methods lead to significant improvements in computational performance.

1. Introduction

The goal of this paper is to illustrate how recently developed solution methods for mixed-integer programming (MIP) scheduling models can be applied to address large-scale problems with a range of complex processing constraints often found in practice. Specifically, we explore four methods: (1) a discrete-time model that employs different time grids for each task/unit, material, and utility to reduce the number of binary variables (Velez and Maravelias, 2013, 2014); (2) a demand-based back-propagation algorithm for the calculation of parameters used to generate tightening constraints in minimization problems (Velez et al., 2013); (3) a time- and inventory-based forward-propagation algorithm to define parameters and tightening constraints in maximization problems (Merchan and Maravelias, submitted for publication) and (4) a reformulation of the basic discrete-time model via the introduction of a new integer variable representing the total number of times each task runs (Velez and Maravelias, 2013).

In this paper, we show how to extend these methods to account for industrially relevant features and then how to combine them to address large-scale instances. We focus on problems in network production environments, that is, environments where a task can consume or produce multiple materials, the output of multiple batches of the same task can be mixed (batch mixing), the output of a single batch can be consumed by multiple downstream batches of the same or different task (batch splitting), and there are recycle streams. While methods (2)–(4) are applicable to both discrete-time and continuous-time models (Merchan et al., 2013; Merchan and Maravelias, 2014), we focus on the former because they are more general, can be readily extended to account for a number of processing constraints, and were recently shown to be computationally superior to continuous-time models for problems in network environments (Sundaramoorthy and Maravelias, 2011).

The paper is structured as follows. In Section 2 we present background material including the basic discrete-time formulation

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

http://dx.doi.org/10.1016/j.ces.2015.05.021

and extensions for variable conversion coefficients, changeovers, material storage in processing units, continuous processes, and buffer tanks. In Section 3, we summarize our solution methods, including a discrete-time multi-grid formulation (Section 3.1), the tightening methods based on demand (Section 3.2) and time/inventory availability with modifications to include processes with variable conversion coefficients (Section 3.3), and a reformulation with an extension for changeovers (Section 3.3). Finally, in Section 4 we describe three large examples and present computational results for different solution methods. We use lowercase italics for indices, uppercase bold letters for sets, uppercase italics for variables, and lowercase Greek letters for parameters.

2. Background

In this section, we present an overview of modeling and solution advances in the area of chemical production scheduling (Section 2.1), present the basic discrete-time state-task network (STN) model (Section 2.2), and then discuss some extensions (Sections 2.4-2.11).

2.1. Literature review

Scheduling problems can be solved using a variety of approaches, including (1) simple dispatching rules; (2) *rigorous* scheduling algorithms (e.g. algorithm of Carlier and Pinson, 1989); (3) *heuristic* scheduling algorithms (e.g. the shifting bottleneck procedure for job-shops Adams et al., 1988); (4) general-purpose metaheuristics (e.g. simulated annealing); (5) constraint programming and the associated constrain propagation algorithms (Baptiste et al., 2001; Hooker, 2002); (6) timed automata and reachability analysis algorithms (Panek et al., 2008; Subbiah et al., 2011); (7) mathematical programming (Mendez et al., 2006; Maravelias, 2012); and (8) hybrid methods combining two or more of the above (Hooker, 2000; Jain and Grossmann, 2001). In this paper, we focus on mathematical programming and specifically MIP methods.

In terms of problems, chemical production scheduling problems can be classified in terms of, among other attributes, the production environment (Maravelias, 2012; Harjunkoski et al., 2014). There are three major environments: (1) sequential, (2) network, and (3) hybrid. In a *sequential* environment it is assumed that all tasks produce/consume a single material and batch mixing and splitting are not allowed for both the input and output materials. In a *network* environment tasks can consume and produce multiple materials, and there are no restrictions in the way input and output materials are handled; i.e. multiple batches of one task can be mixed or material produced by a single batch can be consumed by multiple downstream batches of the same or different tasks. Finally, the term *hybrid* is used to describe processes that are not sequential nor network; e.g. processes where a task consumes and/or produces multiple materials some of which have mixing/splitting restrictions.

The various MIP scheduling models can be grouped primarily in terms of the major entities modeled (Maravelias, 2012). To address problems in sequential environments researchers developed the so-called *batch-based* models where batches are assigned to units and then sequenced to enforce resource constraints (Ku and Karimi, 1988; Pinto and Grossmann, 1995; Cerda et al., 1997; Castro and Grossmann, 2005; Prasad and Maravelias, 2008; Sundaramoorthy and Maravelias, 2008). Problems in network environments were addressed using *material-based* models where materials (and material flows and inventories) are explicitly modeled and tracked over time (Kondili et al., 1993; Pantelides, 1994). Interestingly, sequential environments can be viewed as a special case of the network environment, where (1) all tasks consume/produce a single material and (2) materials are subject to material handling restrictions (no splitting/mixing and no recycling). Based on this insight Maravelias and co-workers developed a modeling framework that enables the representation of problems in all types of production environments, including facilities that consist of subsystems of different types (Sundaramoorthy and Maravelias, 2011; Velez and Maravelias, 2013). Furthermore, material-based models have been extended to account for a number of processing characteristics and constraints (Kelly and Zyngier, 2009; Castro et al., 2011; Gimenez et al., 2009a, 2009b). Thus, since they can be used to address problems in all types of environments and account for various processing constraints, material-based models are the focus of this paper.

Schedule optimization for network processes began with discrete-time models where the time horizon is divided into uniform time intervals (Kondili et al., 1993; Pantelides, 1994; Shah et al., 1993). These models appeared to be intractable for the solvers of the time, so most effort to improve solution times then focused on developing smaller models, primarily models employing continuous time representations, where the location of time points is an optimization decision (Zhang and Sargent, 1996; Schilling and Pantelides, 1996). Continuous-time models have fewer time points and fewer binary variables, which led to the belief that they could be solved faster. Continuous-time models relying both on a single grid common across all units (Mockus and Reklaitis 1999; Castro et al., 2001; Maravelias and Grossmann, 2003; Sundaramoorthy and Karimi, 2005) as well as different grids for each unit (Ierapetritou and Floudas, 1998; Janak et al., 2004; Susarla et al., 2010) were developed to reduce the number of time points even further.

In terms of MIP-based solution methods, researchers in the process systems engineering (PSE) community have proposed: (1) tightening methods including preprocessing algorithms for fixing binary variables (Pinto and Grossmann, 1995; Blomer and Gunther, 2000) and generating valid inequalities (Sundaramoorthy and Maravelias, 2008), as well as the solution of auxiliary LP and MIP models for the generation of valid inequalities (Burkard and Hatzl 2005; Janak and Floudas, 2008); (2) reformulations including variable disaggregation (Sahinidis and Grossmann, 1991; Yee and Shah, 1998) and reformulation-linearization (Janak and Floudas, 2008) techniques; (3) decomposition methods relying on the structure of the network (Papageorgiou and Pantelides, 1996), the hierarchy of decisions (Bassett et al., 1996; Kelly and Zyngier, 2008), the iterative solution of a simpler MIP model (Maravelias and Grossmann, 2003), Lagrangean relaxation and decomposition (Wu and Ierapetritou, 2003; Calfa et al., 2013), and rolling horizon approaches (Dimitriadis et al., 1997; Lin et al., 2002); and (4) algorithmic enhancements including preprocessing algorithms to generate strong valid inequalities (Velez et al., 2013; Velez and Maravelias, 2013), and the use of heuristics (Mendez and Cerda, 2003; Roslof et al., 2001; Kopanos et al., 2010). Furthermore, researchers have proposed decomposition methods that rely on the integration of different solution methods, both for sequential (Jain and Grossmann, 2001; Harjunkoski and Grossmann, 2002; Maravelias, 2006) and network (Maravelias and Grossmann, 2004; Roe et al., 2005) environments. Finally, there have been some attempts to design algorithms for distributed and parallel computing (Subrahmanyam et al., 1996; Ferris et al., 2009; Velez and Maravelias, 2013). However, despite the efforts to develop small and tight MIP models as well as effective solution methods, the solution of large-scale scheduling problems in network environments remained challenging until recently.

2.2. Basic model

The structure of the process network is defined in terms of the following sets: