Computers and Chemical Engineering 33 (2009) 1890–1904

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Polyhedral results for discrete-time production planning MIP formulations for continuous processes

Christos T. Maravelias[a,*], Konstantinos Papalamprou[b]

[a] Department of Chemical and Biological Engineering, University of Wisconsin-Madison, 1415 Engineering Dr., Madison, WI 53706, USA
[b] Operational Research Group, Department of Management, London School of Economics and Political Science, Houghton Street, London WC2A 2AE, UK

ARTICLE INFO

Article history:
Received 4 July 2008
Received in revised form 6 May 2009
Accepted 12 May 2009
Available online 3 June 2009

Keywords:
Scheduling
Production planning
Mixed-integer programming
Polyhedral results
κ-Regularity

ABSTRACT

We derive polyhedral results for discrete-time mixed-integer programming (MIP) formulations for the production planning of multi-stage continuous chemical processes. We express the feasible region of the LP-relaxation as the intersection of two sets. The constraints describing the first set yield the convex hull of its integer points. For the second set, we show that for integral data the constraint matrix is κ-regular, and that the corresponding polyhedron is integral if the length of the planning period is selected appropriately. We use this result to show that for rational data, integer variables can also assume integral values at the vertices of the polyhedron. We also discuss how these results provide insight and can be used to effectively address large-scale problems. Finally, we present computational results for a series of example problems.

1. Introduction

The scheduling and production planning of multi-product facilities has received considerable attention in the process systems engineering (PSE) literature (Shah, 2005; Mendez, Cerda, Grossmann, Harjunkoski, & Fahl, 2006; Sung & Maravelias, 2007; Maravelias & Sung, 2009). The focus of most of the existing approaches is on the development of mixed-integer programming (MIP) formulations that are shown to solve well a class of problems. The quality of such MIP formulations is typically assessed from the computational requirements for the solution of a set of example problems. In general, it can be argued that the development of MIP models for the scheduling of chemical processes has been guided by empirical (computational) rather than theoretical results. This is in sharp contrast with efforts on production planning in the operations research (OR) community, where the development of reformulations and the derivation of theoretical results with respect to the structure and tightness of MIP formulations is the method of choice (Pochet & Wolsey, 2006). The advantage of this more rigorous approach is that these results and reformulations point the way to effective decomposition approaches.

In this paper we follow the latter approach. We study the polyhedral properties of a discrete-time state-task network (STN) formulation (Kondili, Pantelides, & Sargent, 1993) for the production planning of multi-stage multi-product continuous processes. We decompose the problem into two subproblems and develop results regarding the tightness of the two subproblems. The central results concern the length of the time period: we determine the length that yields the tightest possible MIP formulation for the second subproblem.

The paper is structured as follows. In the next section we review concepts and known results on polyhedral theory, total unimodularity, κ-regularity, and decomposition approaches. In Section 3, we formally define the production planning problem we consider in this paper, and present three MIP formulations. We then decompose the proposed formulation into two subproblems and develop polyhedral results for their LP-relaxations. In Section 5, we present a geometric illustration of our results and we discuss how they can be used to solve practical problems. Finally, in Section 6 we present computational results for a series of example problems.

2. Background

2.1. Preliminaries

We will use capital bold letters for sets, lowercase italic letters for indices and scalars, lowercase bold–italic letters for vectors, and uppercase letters for matrices. For the MIP formulations, we use lowercase italic Greek letters to denote parameters and uppercase italic Latin letters for optimization variables. We use small bold–italic letters for the vectors of these parameters and variables. We provide proofs only for new results.

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

0098-1354/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2009.05.015

Nomenclature

Indices/sets

$i \in \mathbf{I}$	Tasks
$j \in \mathbf{J}$	Units
$k \in \mathbf{K}$	Chemicals
$n \in \mathbf{N}$	Planning periods/time points

Subsets

$\mathbf{I}_j$	Tasks that can be carried out in unit j
$\mathbf{I}_{k-}/\mathbf{I}_{k+}$	Tasks consuming/producing chemical k
$\mathbf{K}^{\mathrm{RM/INT/FP}}$	Raw materials/intermediates/final products
$\mathbf{N}^V$	Time points where shipments can be made

Variables

$W_{in} \in \{0, 1\}$	1 if task i is processed in period n
B_{in}	Extent of task i in period n
S_{kn}	Inventory level of chemical k at time point n
$\bar{S}_{kn}$	Generalized inventory of k at time t
U_{kn}	Unmet (backlogged) demand for chemical k in period n
V_{kn}	Shipments of chemical k at time point n

Parameters

β_i	Batchsize of task i
γ_{kn}	Net delivery of chemical k at time point n
Δt	Length of time bucket
ζ_k	Storage capacity for chemical k
η	Planning horizon
θ_k	Holding cost of product k
π_i	Processing cost for task i
ρ_i	Production rate of task i
σ_k	On-hand inventory of chemical k

We will use $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{Q}$, and $\mathbf{R}$ to denote the sets of nonnegative integer (natural), integer, rational and real numbers, respectively, and $\mathbf{Z}_+$ and $\mathbf{R}_+$ to denote the sets of nonnegative integer and real numbers. The set of rows and columns of a matrix A is denoted by $\mathbf{L}$ and $\mathbf{M}$, respectively; the element in row $l \in \mathbf{L}$ and column $m \in \mathbf{M}$ is denoted by a_{lm}. Matrix κA is the matrix obtained by multiplying all its elements by κ; matrix $\kappa^{-1}A = A/\kappa$ is obtained by dividing all elements of A by κ. Vectors and matrices with integer elements are called *integral*. The set of m-dimensional vectors whose elements are integer multiples of $\kappa \in \mathbf{Q}$ is denoted by $\kappa\mathbf{Z}^m$. For an integral vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}}$, the greatest common divisor of $\boldsymbol{x}$, denoted by $gcd(\boldsymbol{x})$, is the greatest integer that divides all elements of $\boldsymbol{x}$. The least common multiple of $\boldsymbol{x}$, denoted by $lcm(\boldsymbol{x})$, is the smallest positive integer that is multiple of all elements of $\boldsymbol{x}$.

In the next four subsections, we present definitions and results on polyhedra and polytopes, total unimodularity, κ-regularity, and decomposition methods. More details can be found in Nemhauser and Wolsey (1988), Cook, Cunningham, Pulleyblank, and Schrijver (1997), Wolsey (1998), Kotnyek (2002), Appa and Kotnyek (2004), and Pochet and Wolsey (2006).

2.2. Polyhedra

Definition 1. A polyhedron $\mathbf{P}$ is the solution set of a finite system of linear inequalities, i.e. $\mathbf{P} = \{\boldsymbol{x}: A\boldsymbol{x} \le \boldsymbol{b}\}$.

Definition 2. A polyhedron $\mathbf{P}$ is a polytope (i.e. a bounded polyhedron) if there exist $\boldsymbol{l}, \boldsymbol{u} \in \mathbf{R}^n$ such that $\boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}$; i.e. $\mathbf{P} = \{\boldsymbol{x}: A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}$ for $\boldsymbol{l}, \boldsymbol{u} \in \mathbf{R}^n\}$.

Since a set of inequalities can be converted into a set of equalities by introducing slack variables, and a set of equalities $A\boldsymbol{x} = \boldsymbol{b}$ can be converted into a set of inequalities ($[A, -A]^{\mathrm{T}}\ \boldsymbol{x} \le [\boldsymbol{b}, -\boldsymbol{b}]^{\mathrm{T}}$), the definitions of polyhedra and polytopes hold also for systems of linear equalities. Hence, a polyhedron $\mathbf{P}$ can be described by a set of linear equalities, $\mathbf{P} = \{\boldsymbol{x}: A\boldsymbol{x} = \boldsymbol{b}\}$, and a polytope can be described by a set of linear equalities and bounding inequalities, $\mathbf{P} = \{\boldsymbol{x}: A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}\}$. This is the description we use in this paper. A polytope can also be described as the convex hull of a finite number of points. Conversely, the convex hull of a finite number of points is a polytope. A vector $\boldsymbol{v}$ of the polyhedron is a *vertex* (extreme point) if and only if it cannot be written as a linear combination of points in $\mathbf{P}\backslash\{\boldsymbol{v}\}$. A polytope $\mathbf{P}$ is the convex hull of its vertices.

It is known that, when finite, the optimal solution of linear programming (LP) models lies at a vertex of the feasible region. Thus, LP models can be solved effectively by *searching* these vertices until an optimality condition (i.e. all reduced costs of non-basic variables are non-positive for a maximization problem) is satisfied. This is the main idea behind the simplex method. Mixed-integer programming (MIP) models, on the contrary, are hard to solve: optimal solutions do not typically lie at the vertices of the feasible region of the LP-relaxation (i.e. the region defined by the constraints of the MIP model and the integrality constraints relaxed).

2.3. Total unimodularity

The standard method for the solution of a MIP problem $\max\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{d}^{\mathrm{T}}\boldsymbol{y}\colon A\boldsymbol{x} + B\boldsymbol{y} = \boldsymbol{b},\ \boldsymbol{x} \in \{0,1\}^{N1},\ \boldsymbol{y} \ge 0\}$ until the 1970s was the *branch-and-bound* algorithm (Land & Doig, 1960). However, an alternative method was suggested by the polyhedral results of Edmonds (1965) and Fulkerson (Fulkerson & Gross, 1965; Ford & Fulkerson, 1962) in the 1960s, where they showed that special classes of integer programming (IP) problems $\max\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}: A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \{0, 1\}^n\}$ can be solved by simply dropping the integrality constraints and solving the LP-relaxation $\max\{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}: A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in [0, 1]^n\}$. In other words, they showed that the polytope $\mathbf{P} = \{\boldsymbol{x}: A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in [0, 1]^n\}$ for these classes of problems is *integral*.

Definition 3. A nonempty polyhedron (polytope) is *integral* if its vertices are integral.

Clearly, if an IP problem has a formulation for which the feasible region of its LP-relaxation (henceforth referred to as the polyhedron or polytope of the problem) is integral, then this problem can be solved effectively. But what are the problems that have integral polyhedra? The famous theorem of Hoffman and Kruskal (1956) provides an answer for the case of integral matrices and right hand side vectors.

Theorem 1. *The polyhedron* $\boldsymbol{P}(A, \boldsymbol{b}) = \{\boldsymbol{x}: A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \ge 0\}$ *is integral for all integral vectors* $\boldsymbol{b} \in \boldsymbol{Z}^m$ *if and only if A is totally unimodular.*

Definition 4. An integral matrix A is totally unimodular (TU) if the determinant of each square submatrix of A is equal to 0, 1, or −1.

Theorem 1 can be extended to problems with bounds:

Proposition 1. *If A is TU and vectors* $\boldsymbol{b}$, $\boldsymbol{b}'$, $\boldsymbol{d}$ *and* $\boldsymbol{d}'$ *are integral, then* $\boldsymbol{P}(\boldsymbol{b}, \boldsymbol{b}', \boldsymbol{d}, \boldsymbol{d}') = \{\boldsymbol{x} \in R_+^n: \boldsymbol{b}' \le A\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{d}' \le \boldsymbol{x} \le \boldsymbol{d}\}$ *is integral if it is not empty.*

A property of TU matrices that we will use later is given by the following proposition:

Proposition 2. *For every non-singular submatrix* R *of a TU matrix,* R^{-1} *is integral.*

An important subclass of TU matrices is the class of network matrices (for a thorough overview the reader is referred to Nemhauser & Wolsey, 1988). Any network matrix is associated