I&EC research
Industrial & Engineering Chemistry Research

Efficient Solution of Enterprise-Wide Optimization Problems Using Nested Stochastic Blockmodeling

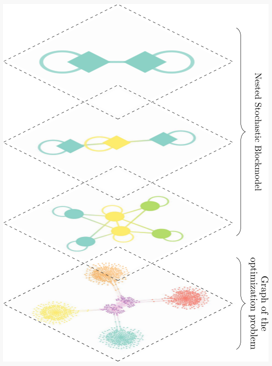
Ilias Mitrai and Prodromos Daoutidis*

Cite This: *Ind. Eng. Chem. Res.* 2021, 60, 14476−14494

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information

ABSTRACT: Enterprise-wide optimization seeks to improve the economic performance of process systems by considering simultaneously decisions at different time scales, resulting in large-scale optimization problems. In this paper, we propose the application of nested stochastic blockmodeling (nSBM) for the decomposition of such optimization problems. This approach allows the identification of the block structure of the problem at different hierarchical levels and the hierarchy itself. We consider problems of integration of scheduling and dynamic optimization and integration of planning, scheduling, and dynamic optimization for illustration. Application of nSBM reveals the multiscale nature of these optimization problems, and the exploitation of the structure of the problem at different hierarchical levels enables efficient solutions.


Nested Stochastic Blockmodel
Graph of the optimization problem

INTRODUCTION

The integration of process operations is considered as a promising avenue to improve the economic performance of process systems.[1,2] This approach considers simultaneously different decision-making problems, resulting in large-scale optimization problems whose solution is a challenging task. The difficulty arises from the inherently nonlinear behavior of most process systems and the different time scales that are involved. For example, supply chain/planning decisions span a time horizon of months, scheduling decisions are made weekly, and control decisions are made in the time scale of seconds or minutes. The integration of these decision layers leads to multiscale optimization problems, which are generally nonscalable and whose monolithic solution can be intractable.

Different approaches have been proposed to reduce the complexity of such problems and improve the computational time. In one approach, the problem is decomposed into distinct steps, where the different problems are solved hierarchically, and the solution of the upper level problem is an input to the lower level problems.[3] In this approach, although the computational time is reduced, the solution can be suboptimal. A second approach to improve the tractability of a problem is to approximate the computationally complex part with a simpler surrogate model.[4] Typically, this approach is used to handle long time horizons and approximate the behavior of nonlinear systems.[3,5−10]

A different approach is to exploit the structure of the full problem and apply decomposition-based solution methods. Typical examples are the application of Lagrangean and augmented Lagrangean relaxation/decomposition,[11−14] Benders and Generalized Benders decomposition,[15−18] and bilevel decomposition.[19,20] Although this approach can reduce the computational time, a decomposition of the problem itself is necessary. The decomposition chosen is typically problem-specific or is based on intuition. In general, we can argue that decomposition-based solution algorithms exploit some block structure in the problem.[21,22] For example, in Benders decomposition, the problem is decomposed into a master problem and a subproblem, where the latter provides information about the effect of the master problem on the solution of the subproblem. This can be considered as a hierarchical structure, since the solution of the subproblem depends on the solution of the master problem. On the contrary, in Lagrangean relaxation/decomposition, the problem is decomposed into subproblems which are coupled through a number of constraints, called complicating constraints. This partition implies a weakly coupled block structure in the problem, where the value of the coupling constraints affects the solution of the subproblems. Despite this direct relation between the block structure and decomposition-based solution algorithms, the block structure of an optimization problem is not always apparent. Therefore, the detection of the underlying structure of a problem is an important first step toward the selection of the most appropriate decomposition-based solution algorithm.

Received: April 26, 2021
Revised: September 3, 2021
Accepted: September 7, 2021
Published: October 4, 2021

ACS Publications

https://doi.org/10.1021/acs.iecr.1c01570

In recent research, we have proposed the application of network science concepts and tools to systematically decompose optimization problems.[23,24] In this approach, the structural coupling among the variables and constraints of an optimization problem is captured through the variable and constraint graphs. In the variable graph, the nodes are the variables of an optimization problem; the edges are the constraints that couple two variables and a weight, which is equal to the number of such constraints, can be assigned to each edge. Similarly, in the constraint graph, the nodes are the constraints of the problem, and the edges are the variables that are present in the constraints. This graph representation of an optimization problem allows the systematic analysis of its structure using tools from network science. The first tool that was employed was community detection.[25] This approach can provide high-quality decompositions, whereby groups (blocks) of variables or constraints are identified with weak interactions (in a statistical sense).[23,26] Centrality analysis can be used in conjunction to reveal the hierarchical relation among the communities.[27] Although this approach can provide decompositions that can reduce the computational time, the main assumption is that the graph and hence the optimization problem has a community structure. In Mitrai et al.,[24] we have proposed the application of stochastic blockmodeling (SBM) and statistical inference as a framework to learn the underlying structure of optimization problems without any a priori assumptions on the structure of the problem. Stochastic blockmodels are random graph generative models that allow the generation of random graphs with arbitrary block structures.[28] Statistical inference allows learning the SBM that generated a given problem and therefore detecting its underlying block structure. The estimated structure can then be used as the basis for the application of decomposition-based solution algorithms, which can reduce the necessary time to obtain a solution or an estimate of its lower and upper bound. However, neither of these approaches (community detection or SBM) can detect the possible hierarchical or multiscale structure of a problem.

In this paper, we propose the application of *nested* stochastic blockmodeling (nSBM) for learning the hierarchical block structure of optimization problems. nSBM describes a nested hierarchy of stochastic blockmodels, where the connections among the nodes depend solely on their block affiliation.[29−31] The parameters of the model, which are estimated through statistical inference,[29,32] reveal the block structure of the problem at different hierarchical levels and the hierarchy itself. We apply this approach to optimization problems that arise in the integration of process operations, and we show that it can indeed learn the hierarchical multiscale structure of these problems and guide the application of decomposition-based solution algorithms.

In the rest of the manuscript, we begin by presenting the nSBM model, the inference problem, and solution approaches. In the next two sections, we apply this approach to the integration of scheduling and dynamic optimization and the integration of planning, scheduling, and dynamic optimization. For both cases, we analyze the structure of the problem and compare the decompositions that are obtained at different hierarchical levels.

■ NESTED STOCHASTIC BLOCKMODELING AND BAYESIAN INFERENCE

In this section, the stochastic model and inference approaches will be presented for the simple nested stochastic blockmodel to keep the notation simple. At the end of the section, we will discuss extensions and variants of this basic model.

Stochastic Blockmodel. Consider an undirected graph $G(V, E)$ with N nodes, M edges ($|V| = N$, $|E| = M$), and adjacency matrix $A \in \mathbb{R}^{N\times N}$, where $A_{ij} = A_{ji}$ is equal to the number of edges between node i and j. We will assume that the nodes are assigned into B blocks and will define a partition vector $b \in \mathbb{R}^{N}$, where $b_i \in \{1, ..., B\}$ denotes the group membership of node i. We also define the matrix $\omega \in \mathbb{R}^{B\times B}$, where ω_{rs} is equal to the number of edges between the nodes that belong in block r and the nodes that belong in block s, and ω_{rr} is equal to twice the number of edges between the nodes in block r. The entries of the ω matrix are equal to

$$\omega_{rs} = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}\delta_{b_i r}\delta_{b_j s} \quad \forall\ r = 1, ..., B,\ s = 1, ... B \tag{1}$$

where δ is the Kronecker delta. Finally, we define $n \in \mathbb{R}^{B}$, where n_r is equal to the number of nodes in block r and is equal to

$$n_r = \sum_{i=1}^{N} \delta_{b_i r} \quad \forall\ r = 1 ,.., B \tag{2}$$

We note that for a given graph with adjacency matrix A, both ω and n depend on the partition of the nodes b.

For a simple graph ($A_{ij} \in \{0, 1\}$), given a number of nodes N and a partition b into B blocks, different graphs with the same ω matrix can be generated, and all of the graphs are equally probable. This leads to an ensemble of graphs, where the total number of different graphs is equal to[33]

$$\Omega(\omega, b) = \prod_{s=1,r=1,r\geq s}^{B} \Omega_{rs} \tag{3}$$

where

$$\Omega_{rs} = \binom{n_r n_s}{\omega_{rs}},\quad \Omega_{rr} = \binom{\binom{n_r}{2}}{\frac{\omega_{rr}}{2}} \tag{4}$$

The first expression above is equal to all of the possible ways to select ω_{rs} edges from $n_r n_s$ ($n_r n_s$ is equal to all of the possible edges between all of the nodes in block r and all of the nodes in block s). Similarly, the second expression is equal to the number of ways to assign $\omega_{rr}/2$ edges between the nodes in block r. Since all of these graphs are equally probable, the probability to observe a graph G given a partition b is equal to

$$P(G|b) = \frac{1}{\Omega(\omega, b)} \tag{5}$$

and the entropy of the ensemble is equal to

$$S_g(\omega, b) = \ln \Omega(\omega, b) \tag{6}$$

For a multigraph, defined as a graph with multiple edges between two nodes ($A_{ij} \in \mathbb{Z}_+$), the different ways to create a graph with N nodes and a partition b into B blocks are equal to[33]

$$\Omega_m(\omega, b) = \prod_{r=1, s=1, r \geq s}^{B} \Omega_{rs}^m \tag{7}$$

where

$$\Omega_{rs}^m = \left(\begin{matrix} n_r n_s + \omega_{rs} - 1 \\ \omega_{rs} \end{matrix} \right), \quad \Omega_{rr}^m = \left(\begin{matrix} \dfrac{n_r n_r}{2} + \dfrac{\omega_{rr}}{2} - 1 \\ \dfrac{\omega_{rr}}{2} \end{matrix} \right) \tag{8}$$

The first expression is equal to the number of ω_{rs} combinations with repetition from a set with size $n_r n_s$. Similarly, as in the case of a simple graph, all multigraphs are equally probable, the probability to observe a multigraph is $1/\Omega_m(\omega, b)$, and the entropy is

$$S_m(\omega, b) = \ln \Omega_m(\omega, b) \tag{9}$$

Nested Stochastic Blockmodel. The nSBM is based on the idea that for a given graph $G(V, E)$ (simple or multigraph), a partition b into B blocks leads to an ω matrix, whose entries are positive integer values ($\omega_{rs} \in \mathbb{Z}_+$). Therefore, this ω matrix can be considered as the adjacency matrix of a new multigraph G' with B nodes and E edges, i.e., $\omega = A'$, where A' is the adjacency matrix of multigraph G'. Similarly, the nodes in this new multigraph can be partitioned into B' blocks, leading in turn to a new ω' matrix, which can be considered as the adjacency matrix of a new multigraph G'' with B' nodes. This procedure can be continued until one block is left, leading to a nested sequence of stochastic blockmodels, where the first level simple graph is the observed network, and the multigraphs in the other levels form the nested model.[29] An example of a nested stochastic blockmodel with three levels is presented in Figure 1. The colors used to denote the blocks in a specific level do not have any association with the colors in the other levels.

Let us assume that $l = L$ levels exist and $l = 0$ corresponds to the observed network. The number of nodes in level l multigraph is B_l ($B_l \leq B_{l-1}$), and the number of edges is E. Based on the definition of the nSBM, the last level $l = L$ has one block ($B_L = 1$). We define $\omega^l \in \mathbb{R}^{B_l \times B_l}$, where ω_{rs}^l is equal to the number of edges between the nodes in block r and s in level l. We also define $n^l \in \mathbb{R}^{B_l}$, where n_r^l is the number of nodes in block r in level l, and $b^l \in \mathbb{R}^{B_l}$ is the partition of the nodes in level l into B_l blocks. Given these definitions, the following equations hold:

$$B_{l-1} = \sum_{r=1}^{B_l} n_r^l, \quad E = \sum_{r=1}^{B_l} \sum_{s=1}^{B_l} \omega_{rs}^l / 2 \tag{10}$$

$$\omega_{rs}^l = \sum_{i=1}^{B_{l-1}} \sum_{j=1}^{B_{l-1}} A_{ij}^l \delta_{b_i^l r} \delta_{b_j^l s} \quad \forall\ r = 1, \ldots, B_{l-1},\ s = 1, \ldots B_{l-1} \tag{11}$$

Based on these equations, the total number of edges is the same in all of the levels, and the number of nodes decreases. The entropy of the nested SBM is equal to

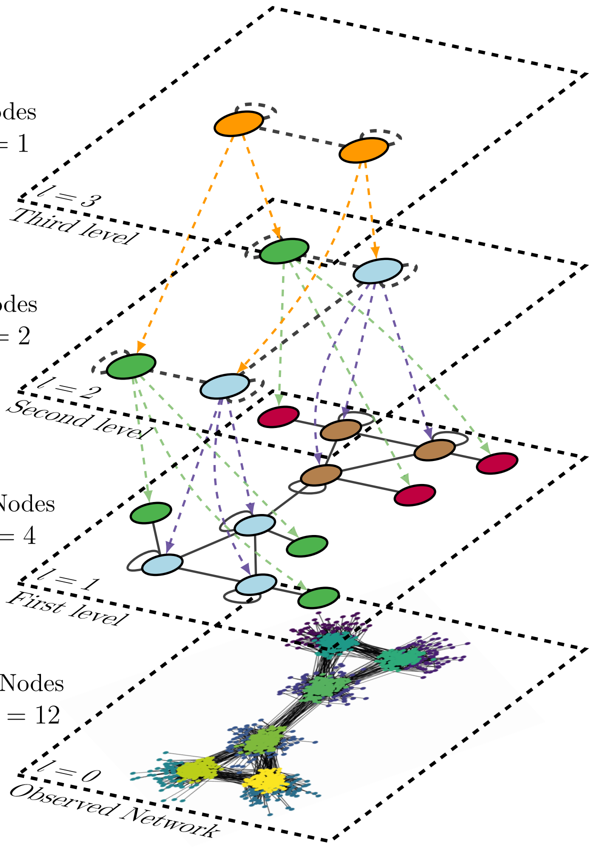


Figure 1. Example of a nested SBM (reproduced/adapted with permission from Peixoto.[30] Copyright 2020 Wiley). The observed graph has $N = 1273$ nodes, $E = 8309$ edges, and the nodes of the observed network are partitioned into $B_0 = 12$ blocks. The hollow circles denote self-edges.

$$S_n = S_g(\omega^0, b^0) + \sum_{l=1}^{L} S_m(\omega^l, b^l) \tag{12}$$

where S_g and S_m are given by eqs 6 and 9. The first term is the entropy of the observed simple graph, and the second term accounts for the entropies of the multigraphs in each level l. Since all graphs at all levels are equally probable, the probability to observe a nested SBM is equal to

$$P = \frac{1}{\Omega(\omega^0, b^0) \prod_{l=1}^{L} \Omega_m^l(\omega^l, b^l)} \tag{13}$$

Inference Approach. Given a graph $G(V, E)$, the goal is to infer the block partition and ω matrix for all of the levels that best fit the data (the observed graph). Two approaches can be followed to solve the problem, maximum likelihood estimation (MLE) and Bayesian inference. In this work, we will focus on the second approach, which involves the estimation of the posterior probability of the observed network. If we assume that one level exists ($L = 1$), then from Bayes' rule

$$P(b|A) = \frac{P(A|b)P(b)}{P(A)} \tag{14}$$

where

$$P(A|b) = P(A|\omega, b)P(\omega|b)\ ,\ \ P(A) = \sum_{b} P(A|b)P(b) \tag{15}$$

In Bayesian inference, the goal is to find the probability distribution $P(b|A)$. Based on this distribution, different partitions can be sampled, and the partition that maximizes the probability can be found. The numerator in the above expression can be written as

$$P(A|b)P(b) = P(A|\omega, b)P(\omega|b)P(b) = P(A|\omega, b)P(\omega, b) \tag{16}$$

Therefore, the original task of maximizing $P(A|b)$ is equivalent to maximizing $P(b|A)$ or minimizing $-P(b|A)$, which is equal to

$$\text{minimize}_{b,\omega} - \log_2 P(A|\omega, b) - \log_2 P(\omega, b) \tag{17}$$

This objective function has an information theoretical interpretation. The first term is the amount of bits necessary to encode the observed data, and the second term is the amount of bits necessary to encode the parameters of the model. Hence, the Bayesian inference approach lends itself naturally to the usage of the description length $\Sigma = -\log_2 P(A|\omega, b) - \log_2 P(\omega, b)$ as the objective function and avoids overfitting. An increase in the number of blocks leads to a reduction in the first term, but the model is more complex, leading to an increase in the second term.

For the nested SBM case, the goal is to estimate or maximize $P(\{\mathrm{b}^l\}_{l=1}^{L}|A)$, which is equal to[32]

$$P(\{b^l\}_{l=1}^{L}|A) = \frac{P(A|\{b^l\}_{l=1}^{L})P(\{b^l\}_{l=1}^{L})}{P(A)} \tag{18}$$

Similar to the case of a single level, the numerator can be written as

$$P(A|\{\omega^l\}_{l=1}^{L}, \{b^l\}_{l=1}^{L})\ P(\{\omega^l\}_{l=1}^{L}, \{b^l\}_{l=1}^{L}) \tag{19}$$

and the inference problem is

$$\begin{aligned}&\text{minimize}_{\{b^l\}_{l=1}^{L},\{\omega^l\}_{l=1}^{L}}(-\log_2 P(A|\{\omega^l\}_{l=1}^{L}, \{b^l\}_{l=1}^{L})\\ &\quad - \log_2 P(\{\omega^l\}_{l=1}^{L}, \{b^l\}_{l=1}^{L}))\end{aligned} \tag{20}$$

In this paper, we will use the microcanonical ensemble approach to define the prior distribution.[32] The prior for the partition in all levels, $P(\{b^l\}_{l=1}^{B^l})$, is equal to

$$P(\{b^l\}_{l=1}^{L}) = \prod_{l=1}^{L} P(b^l) \tag{21}$$

We can assume that the size of the different blocks depends on the number of blocks. Therefore, given the sizes of the different blocks, n^l, the probability to observe b^l is equal to

$$P(b^l|n^l) = \frac{\prod_{r=1}^{B_l} n_r^l!}{B_{l-1}!} \tag{22}$$

Additionally, the different ways to assign the nodes in level l into B_l nonempty blocks is $\binom{B_{l-1}-1}{B_l-1}$. Hence, the probability to obtain n^l given B^l is

$$P(n^l|B_l) = \binom{B_{l-1}-1}{B_l-1}^{-1} \tag{23}$$

Finally, we can assume that the probability of having B^l blocks is $P(B^l) = 1/B_{l-1}$, where B_{l-1} is the maximum number of blocks in level l, i.e., every node is assigned into one block. Overall, the prior probability of the block assignments in level l is

$$\begin{aligned}P(b^l) &= P(b^l|n^l)P(n^l|B_l)P(B_l)\\ &= \frac{\prod_{r=1}^{B_l} n_r^l!}{B_{l-1}!}\binom{B_{l-1}-1}{B_l-1}^{-1} B_{l-1}^{-1}\end{aligned} \tag{24}$$

The prior for $P(\{\omega^l\}_{l=1}^{L}|\{b^l\}_{l=1}^{L})$ is equal to

$$\begin{aligned}&P(\{\omega^l\}_{l=1}^{L}|\{b^l\}_{l=1}^{L})\\ &\quad= \prod_{l=1}^{L} P(\omega^l|\omega^{l+1}, b^l)\\ &\quad= \prod_{l=1}^{L} P(A^{l+1}|\omega^{l+1}, b^l)\end{aligned} \tag{25}$$

where

$$\begin{aligned}P(\omega^l|\omega^{l+1}, b^l) &= \prod_{r<s}\binom{n_r^l n_s^l + \omega_{rs}^{l+1} - 1}{\omega_{rs}^{l+1}}^{-1}\\ &\quad\times \prod_{r}\binom{n_r^l(n_r^l + 1)/2 + \omega_{rs}^{l+1}/2 - 1}{\omega_{rs}^{l+1}/2}^{-1}\end{aligned} \tag{26}$$

Based on the above equation, the probability $P(\omega^l|\omega^{l+1}, b^l)$ to observe a multigraph in level l+1 with adjacency matrix $A^{l+1} = \omega^l$, depends on b^l, ω^{l+1}. b^l dictates the number of nodes in level l+1, and ω^{l+1} dictates the connection pattern among the nodes. Note that the expression for $P(\omega^l|\omega^{l+1}, b^l)$ is equal to $1/\Omega_m^l$. Given the above prior distributions, the posterior is estimated using a Markov chain Monte Carlo approach as described in Peixoto.[30,32] In this method, at each level, node move proposals are made based on the block affiliation of the neighbor nodes using a Metropolis−Hastings criterion, which guarantees ergodicity, i.e., all possible partitions are possible.

The above formulation and inference approach for the basic nSBM model can be extended to account for more general models. Specifically, we first mention the degree-corrected nSBM.[30,32] In the simple nSBM, nodes with high degree tend to be assigned to the same block. This issue is resolved using the degree-corrected version (DC-nSBM), where the prior of the degree distribution depends on some hyperparameters, which are estimated from the data, i.e., observed network.

The second variant of the nSBM, called weighted nSBM,[34] accounts for edge weights, which denote the strength of coupling among the nodes. We define the observed weights as x, where x_{ij} is the weight of an edge between node i and j, and the probability distribution of the weights depends solely on the group membership of nodes i and j. If we assume for simplicity that only one level exists, then given a number of nodes n, a partition b and γ, and a parameter that governs the sampling of the weights, a graph with adjacency matrix A and weights x can be generated, and the probability to observe it is equal to

$$P(A, x|\theta, \gamma, b) = P(x|A, \gamma, b)P(A|\theta, b) \tag{27}$$

where θ are the parameters of the model. The partition that maximizes the probability to observe a partition b, given a graph A and weights x, is

$$P(b|A, x) = \frac{P(A, x|b)P(b)}{P(A, x)} \tag{28}$$

The posterior distribution can be estimated following a Bayesian inference approach. Based on the data, different models can be used for the weights. We refer the reader to Peixoto[34] for a detailed explanation of the weighted nSBM model and the inference approach. We must note that although the number of blocks in the observed network B_0 can be inferred from the data, it can also be used as a tunable hyperparameter.

In the subsequent sections, we illustrate the application of this modeling and inference framework to two different classes of enterprise-wide optimization problems.

INTEGRATION OF SCHEDULING AND DYNAMIC OPTIMIZATION

Optimization Model. First, we will consider the integration of scheduling and dynamic optimization for continuous parallel lines. The problem formulation is based on Flores-Tlacuahuac and Grossmann.[35] We assume that N_p ($\mathcal{I}_p = \{1, ..., N_p\}$) products must be produced, N_l ($\mathcal{I}_l = \{1, ..., N_l\}$) production lines are available, and the time horizon in each line, H_l, is divided into N_s ($\mathcal{I}_s = \{1, ..., N_s\}$) slots.

Scheduling Model. First, we define the variable $Y_{ikl} \in \{0, 1\}$, which is one, if product i is produced at slot k in line l and zero, otherwise. In each slot and line, only one product can be produced, and each product is produced at least once. These constraints are modeled through the following equations

$$\sum_{i=1}^{N_p} Y_{ikl} = 1 \quad \forall \;\; k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$\sum_{l=1}^{N_l}\sum_{k=1}^{N_s} Y_{ikl} \geq 1 \quad \forall \;\; i \in \mathcal{I}_p \tag{29}$$

Additionally, the variable $z_{ijkl} \in [0, 1]$ is introduced to denote a transition from product i to product j at slot k and line l, which depends on the value of the Y_{ikl} variable as follows

$$\sum_{i=1}^{N_p} z_{ijkl} = Y_{jkl} \quad \forall \;\; j \in \mathcal{I}_p, k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$\sum_{j=1}^{N_p} z_{ijkl} = Y_{ik-1l} \quad \forall \;\; i \in \mathcal{I}_p, k \in \mathcal{I}_s, k \neq 1, l \in \mathcal{I}_l \tag{30}$$

Each line is composed from N_s slots, and the total production time of product i in slot k and line l is t_{ikl}^{prod}. This time is the sum of the production time and the transition time θ_{kl}^t. The timing constraints are the following

$$t_{ikl}^{prod} \leq t_{max} Y_{ikl} \quad \forall \;\; i \in \mathcal{I}_p, k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$H_l = \sum_{i=1}^{N_p}\sum_{k=1}^{N_s} t_{ikl}^{prod} \quad \forall \; l \in \mathcal{I}_l \tag{31}$$

The amount of product i produced in slot k at line l is given by W_{ikl}, and the production rate is r_{il}. The amount of product i that is produced must satisfy the demand (d_i). The constraints are the following

$$W_{ikl} = r_{il}(t_{ikl}^{prod} - \theta_{kl}^t Y_{ikl}) \quad \forall \;\; i \in \mathcal{I}_p, k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$\sum_{k=1}^{N_s}\sum_{l=1}^{N_l} \frac{W_{ikl}}{H_l} = d_i \quad \forall \;\; i \in \mathcal{I}_p \tag{32}$$

The objective of the scheduling model is to minimize the cost, which consists of the production, transition, and storage cost and is given by the following equation

$$\sum_{i=1}^{N_p}\sum_{k=1}^{N_s}\sum_{l=1}^{N_l}\left(C_{il}^p \frac{t_{ikl}^{prod}}{H_l} + c_{il}^{stor} W_{ikl} \right) + \sum_{i=1}^{N_p}\sum_{j=1}^{N_p}\sum_{k=1}^{N_s}\sum_{l=1}^{N_l} c_{ijl}^{trans}\theta_{kl}^t \frac{z_{ijkl}}{H_l} \tag{33}$$

where c_{il}^p is the production cost of product i at line l, c_{ij}^{trans} is the transition cost between product i and j, and c_{il}^{stor} is the inventory cost of product i in line l. These parameters are calculated using the following equations

$$c_{il}^p = c_i^{prod} r_{il} \;, \quad c_{ijl}^{trans} = 0.9 c_i^{prod} r_{il} \;, \quad c_{il}^{stor} = 0.5(1 - d_i/r_{il})c_i^{inv} \tag{34}$$

where c_i^{prod} is the production cost of product i, and c_i^{inv} is the inventory holding cost.

Dynamic Model. We will assume that the dynamic behavior of the system for each line l can be modeled by a general system of differential equations

$$\dot{x}_l^n = F_l(x_l^n, u_l^n) \quad \forall \; l \in \mathcal{I}_l \tag{35}$$

where x_l^n is the value of state n at line l. The dynamic model is discretized using collocation on finite elements and the discretized equations are (N_{fe}, N_{cp} is the number of finite elements and collocation points respectively, and $\mathcal{I}_f = \{1, ..., N_{fe}\}$, $\mathcal{I}_c = \{1, ..., N_{cp}\}$)

$$x_{fckl}^n = x0_{fkl}^n + h_{kl}^{fe} \sum_{m=1}^{N_{cp}} \Omega_{mc} \dot{x}_{fmkl}^n$$
$$\forall \; n, f \in \mathcal{I}_f, c \in \mathcal{I}_c, k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$x0_{fkl}^n = x0_{f-1kl}^n + h_{kl}^{fe} \sum_{m=1}^{N_{cp}} \Omega_{mc} \dot{x}_{f-1,mkl}^n$$
$$\forall \; n, f \in \mathcal{I}_f, f \geq 2, c \in \mathcal{I}_c, k \in \mathcal{I}_s, l \in \mathcal{I}_l$$
$$\dot{x}_{fckl}^n = F^n(x_{fckl}^n, u_{fckl}^m) \quad \forall \; n, f \in \mathcal{I}_f, c \in \mathcal{I}_c, k \in \mathcal{I}_s, l \in \mathcal{I}_l \tag{36}$$

$$h_{kl}^{fe} = \frac{\theta_{kl}^t}{N_{fe}} \quad \forall \; k \in \mathcal{I}_s, l \in \mathcal{I}_l \tag{37}$$

Table 1. Steady-State Conditions, Production and Inventory Cost for All of the Products, a_u = 0.01

product	c^{ss} (mol/L)	Q^{ss} (L/h)	production cost ($/kg)	inventory cost ($/kgh)	production rate (kg/h) line 1	production rate (kg/hr) line 2	demand rate (kg/h)
1	0.24	200	120	5	40	70	18
2	0.2	100	150	5.5	80	65	14
3	0.3032	400	130	7.8	278.8	300	17
4	0.393	1000	125	9	607	500	16

Finally, we define t^d_{fckl}, which is equal to the time at finite element f, collocation point c at slot k in line l, and the following equation holds

$$t^d_{fckl} = h^{fe}_{kl}(f - 1 + \gamma_c) \quad \forall\, f \in \mathcal{I}_f,\, c \in \mathcal{I}_c,\, k \in \mathcal{I}_s,\, l \in \mathcal{I}_l \tag{38}$$

where γ_c is the root of the Lagrange orthogonal polynomial at collocation point c. The dynamic model is used to find the optimal transition profiles for the states and the manipulated variables of the problem. The objective function for a transition from (x_0, u_0) to (x_f, u_f) is given by

$$\int_0^{t_f} (u - u_f)^2 dt \approx \frac{1}{N_{fe}} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} t^d_{fckl} \Lambda_{c,N_{cp}} (u_{fc} - u_f)^2 \tag{39}$$

where γ_c is the Radau root at collocation point c, and Λ is the collocation matrix.

Integrated Problem. The steady-state values of state n and manipulated variable m for product i *are* x^{ss}_{ni} and u^{ss}_{mi}, respectively. Next, we define x^{in}_{nkl}, x^{end}_{nkl}, u^{in}_{mkl}, *and* u^{end}_{mkl}, which are the values of state n and manipulated variable m at the beginning and end of slot k at line l. These variables are related to the scheduling variables through the following constraints

$$x^{in}_{nkl} = \sum_{i=1}^{N_p} x^{ss}_{ni} Y_{ik-1l} \quad \forall\, n,\, k \in \mathcal{I}_s,\, k > 1,\, l \in \mathcal{I}_l$$

$$x^{in}_{n1l} = \sum_{i=1}^{N_p} x^{ss}_{ni} Y_{iN_sl} \quad \forall\, n,\, l \in \mathcal{I}_l$$

$$u^{in}_{mkl} = \sum_{i=1}^{N_p} u^{ss}_{mi} Y_{ik-1l} \quad \forall\, m,\, k \in \mathcal{I}_s,\, k > 1,\, l \in \mathcal{I}_l$$

$$u^{in}_{m1l} = \sum_{i=1}^{N_p} u^{ss}_{mi} Y_{iN_sl} \quad \forall\, m,\, l \in \mathcal{I}_l$$

$$x^{end}_{nkl} = \sum_{i=1}^{N_p} x^{ss}_{ni} Y_{ikl} \quad \forall\, n,\, k \in \mathcal{I}_s,\, l \in \mathcal{I}_l$$

$$u^{end}_{mkl} = \sum_{i=1}^{N_p} u^{ss}_{mi} Y_{ikl} \quad \forall\, m,\, k \in \mathcal{I}_s,\, l \in \mathcal{I}_l \tag{40}$$

$$x0^n_{1kl} = x^{in}_{kl} \quad \forall\, k,\, l$$

$$x^n_{N_{fe}N_{cp}kl} = x^{end}_{kl} \quad \forall\, n,\, k,\, l$$

$$u^m_{11kl} = u^{in}_{kl} \quad \forall\, n,\, k,\, l$$

$$u^m_{N_{fe}N_{cp}kl} = u^{end}_{kl} \quad \forall\, n,\, k,\, l \tag{41}$$

The objective of the integrated problem is to minimize the cost, and the optimization problem is the following (a_u is a weight coefficient)

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N_p} \sum_{k=1}^{N_s} \sum_{l=1}^{N_l} c^p_{il} \frac{W_{ikl}}{H_l} + \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \sum_{j=1}^{N_s} \sum_{l=1}^{N_l} c^{trans}_{ij} \theta^t_{kl} \frac{Z_{ijkl}}{H_l} \\ & + \sum_{i=1}^{N_p} \sum_{k=1}^{N_s} \sum_{l=1}^{N_l} c^{stor}_{il} W_{ikl} \\ & + a_u \sum_{l=1}^{N_l} \sum_{k=1}^{N_s} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} \frac{t^d_{fckl}}{N_{fe}} \Lambda_{c,N_{cp}} (u_{fckl} - u^{end}_{kl})^2 \\ & \text{subject to eqs } 29,\ 30,\ 31,\ 32,\ 36,\ 37\ 38,\ 40,\ 41 \end{aligned} \tag{42}$$

Application of Nested Stochastic Blockmodeling. In this case study, we will assume that four products must be produced in two identical isothermal continuous stirred reactors (2 lines) with two slots. The dynamic behavior of the reactors is modeled by the following equation

$$\frac{dc}{dt} = \frac{Q}{V}(c_{feed} - c(t)) + kc(t)^3 \tag{43}$$

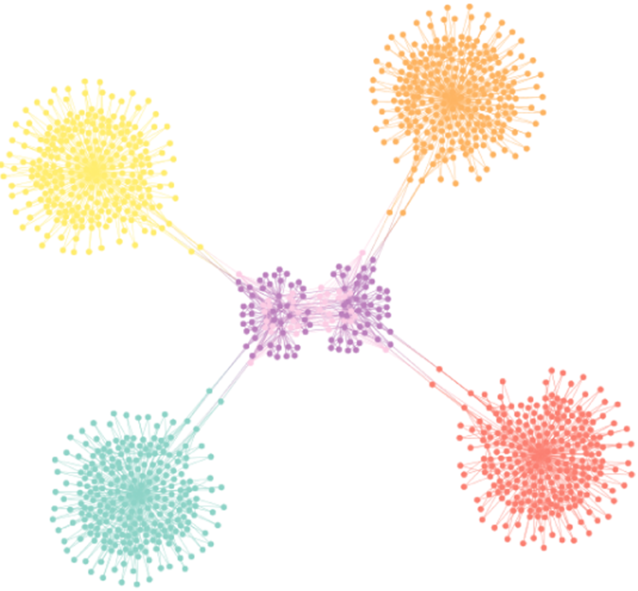


Figure 2. Partition of the variable graph using nested stochastic blockmodeling with a maximum number of blocks equal to 6. The nodes with purple color are Y_{ikl}, z_{ijkl}, x^{in}_{kl}, x^{end}_{kl}, u^{in}_{kl}, and u^{end}_{kl}; the nodes with pink color are θ^t_{kl}, W_{ikl}, H_l, t^{prod}_{ikl}; and the nodes with the other colors correspond to the variables for a slot and line x_{nfckl}, u_{mfckl}, t^d_{fckl}, h^{fe}_{kl}.

where c is the concentration of the reactant, Q is the inlet flow rate (manipulated variable), and $V = 5000$ L, $c_{feed} = 1$ mol/L, $k = 2$ L^2/(h mol^2) are the volume, inlet concentration, and reaction constant parameter. The economic data of the problem are presented in Table 1. First, we analyze the structure of the variable graph, solving the inference problem using Bayesian inference in graph-tool.[36] We present the results for $B_0 = 6$ blocks because this resulted in a decomposition, which is suitable for the application of a decomposition-based solution algorithm and makes intuitive sense. As discussed earlier, in the variable graph, the nodes are the variables of the problem, and an edge corresponds to the constraints that couple two variables. Each edge has a weight which is equal to the number of coupling constraints. Since these weights are positive integer values, we assume that the weights in the nSBM model follow a discrete geometric distribution (see Remark 3 for additional discussion on this assumption). The observed network (variable graph) is decomposed into 6 blocks (Figure 2), and the ω matrix for the partition of the observed graph (ω_0) is

$$\omega_0 = \begin{bmatrix} 336 & 48 & 4 & 4 & 4 & 4 \\ 48 & 177 & 1 & 1 & 1 & 1 \\ 4 & 1 & 756 & 0 & 0 & 0 \\ 4 & 1 & 0 & 756 & 0 & 0 \\ 4 & 1 & 0 & 0 & 756 & 0 \\ 4 & 1 & 0 & 0 & 0 & 756 \end{bmatrix},$$

where the colors in the entries correspond to the graph shown in Figure 2.

Based on the structure of this matrix, the observed graph has a hybrid multicore community structure. The two blocks in the center of the graph form the two cores and correspond to the scheduling variables. The purple block contains variables Y_{ikl}, z_{ijkl}, x_{kl}^{in}, x_{kl}^{end}, u_{kl}^{in}, and u_{kl}^{end}, and the pink block contains the other scheduling variables. The variables that are related to the dynamic behavior of the system (x_{fckl}, u_{fckl}, t_{fckl}^{d} h_{kl}^{fe}) for each slot and line are assigned in the other blocks. These blocks can be considered as communities, since the variables (nodes) in a block are highly coupled with the other variables in the block and loosely coupled with the variables in the two blocks in the core. This structure leads to the nested or double L shape of the ω_0 matrix.

Based on the inference results, three levels are identified (Figure 3) in the nSBM model. The graph in the first level is partitioned into three blocks. This partition reveals a core−periphery structure, where the two blocks in the middle correspond to the scheduling variables, and the other nodes correspond to the variables for the dynamic optimization problem in each slot and line. This structure is evident in the L shape of the ω_1 matrix

$$\omega_1 = \begin{bmatrix} 561 & 10 & 10 \\ 10 & 1512 & 0 \\ 10 & 0 & 1512 \end{bmatrix}$$

Although the nodes for the four dynamic optimization problems are partitioned into two blocks (blue and green nodes), the variables are decoupled since the nodes are not connected directly, i.e., there does not exist an edge between these nodes. This partition is a "coarser" partition of that of the observed graph. The multigraph in the second level is partitioned into two blocks. The yellow square node contains the scheduling variables, and each blue node contains the variables associated with the dynamic optimization for each line. Finally, in the third level, the nodes are assigned in the same block. One node contains the scheduling and the other node, the dynamic optimization variables. Based on these results, we can argue that the estimated nSBM model indeed reveals the multiscale nature of the problem. The partition in the third level shows that two sets of variables exist, scheduling and dynamic optimization ones. In the second level, the dynamic optimization variables are decomposed further into lines, and in the first level, into slots and lines. Finally, the partition of the variables graph (Level 0) reveals the complex interaction among the different variables.

We note that application of community detection and centrality analysis as in Mitrai and Daoutidis[27] to this problem decomposes the variable graph into five communities, and a hierarchy among the communities is identified. The scheduling variables are assigned into one community, and the variables for the dynamic optimization problems for each slot and line are assigned into the other communities. In this partition, all of the variables in the scheduling problem are assigned in the same block, i.e., community. Therefore, community detection and centrality analysis cannot identify the hybrid multicore community structure of the variable graph. This highlights the ability of nSBM to identify the true structure of the problem.

Based on the structure of the ω_0, ω_1 matrices, different decomposition-based solution approaches can be proposed. The core−periphery structure in the first or second level can be used as the basis for the application of Generalized Benders decomposition (GBD),[16] where the variables in the core (nodes with yellow color in the multigraph of the first or second level as shown in Figure 3) are assigned in the master problem and the variables in the periphery, in the subproblem. The hybrid multicore community structure of the observed graph can be used as the basis for the application of nested GBD. The original problem is first decomposed into a master and subproblem. The variables in the two blocks in the core are assigned in the master problem, and the other variables are assigned in the subproblem. The master problem is decomposed further based on the two-block partition of the scheduling variables.

Application of Generalized Benders Decomposition Based on the Structure of the Level 1 Variable Graph. Based on the structure of the ω_1 matrix, we apply Generalized Benders decomposition (GBD). The detailed explanation of the application of GBD based on this partition of the variable graph can be found in Mitrai et al.[24] The scheduling variables (yellow nodes in the first level multigraph, Figure 3) are assigned in the master problem, and the variables associated with the dynamic optimization problem are assigned in the subproblem. The edges that couple nodes that belong to different blocks correspond to constraints that couple variables that belong in the master and subproblem. These constraints are assigned in the subproblem, and the master variables present are the

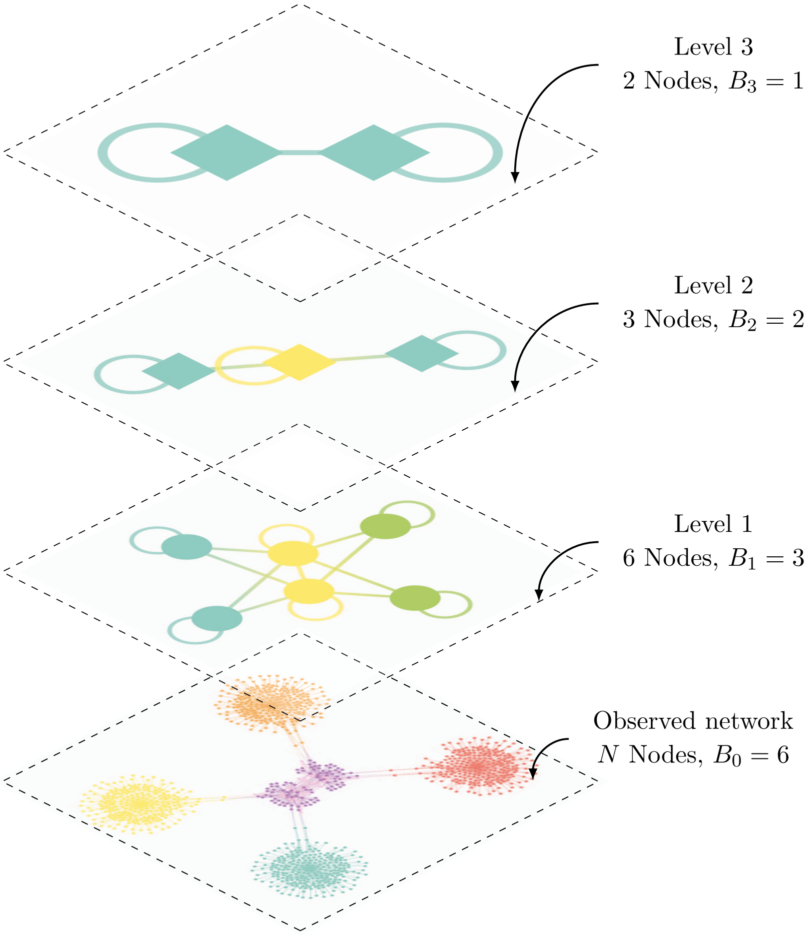


Figure 3. Inferred nSBM model of the variable graph for the integrated scheduling and dynamic optimization problem. The hollow cycles indicate self-edges.

complicating variables. For this case study, these variables are x_{kl}^{in}, x_{kl}^{end}, u_{kl}^{in}, u_{kl}^{end}, and θ_{kl}^{t}. Given this decomposition, the variables are decomposed into three sets. The first set contains the scheduling variables that do not affect directly the subproblem ($s_1 = \{Y_{ikl}, z_{ijkl}, t_{ikl}^{prod}, H_l\}$), the second set contains the variables for the dynamic optimization problem from each slot and line ($s_2 = \{x_{fckl}, u_{fckl}, t_{fckl}^{d}, h_{kl}^{fe}\}$), and the last set contains the complicating variables ($s_3 = \{x_{kl}^{in}, x_{kl}^{end}, u_{kl}^{in}, u_{kl}^{end}, \theta_{kl}^{t}\}$). The subproblem is solved by fixing the shared variables and is equal to

$$\begin{aligned}
&\text{minimize} \quad a_u \sum_{l=1}^{N_l} \sum_{k=1}^{N_s} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} \frac{t_{fckl}^{d}}{N_{fe}} \Lambda_{c,N_{cp}} (u_{fckl} - u_{kl}^{end})^2 \\
&\text{subject to} \quad g_{dyn} \leq 0 \text{ (Eq. 36, 38, 37)} \\
&\qquad x_{kl}^{in} = \bar{x}_{kl}^{in} \quad \forall\, k \in \mathcal{I}_s, l \in \mathcal{I}_l \quad : \quad \lambda_{kl}^{1} \\
&\qquad x_{kl}^{end} = \bar{x}_{kl}^{end} \quad \forall\, k \in \mathcal{I}_s, l \in \mathcal{I}_l \quad : \quad \lambda_{kl}^{2} \\
&\qquad u_{kl}^{in} = \bar{u}_{kl}^{in} \quad \forall\, k \in \mathcal{I}_s, l \in \mathcal{I}_l \quad : \quad \lambda_{kl}^{3} \\
&\qquad u_{kl}^{end} = \bar{u}_{kl}^{end} \quad \forall\, k \in \mathcal{I}_s, l \in \mathcal{I}_l \quad : \quad \lambda_{kl}^{4} \\
&\qquad \theta_{kl}^{t} = \bar{\theta}_{kl}^{t} \quad \forall\, k \in \mathcal{I}_s, l \in \mathcal{I}_l \quad : \quad \lambda_{kl}^{5}
\end{aligned} \tag{44}$$

where the bar denotes that the corresponding variable is fixed, and λ is the Lagrangean multiplier. This problem corresponds to the dynamic optimization problems for every slot and line. Hence, different problems can be solved independently, and their solutions depend on the values of the complicating variables (x_{kl}^{in}, x_{kl}^{end}, u_{kl}^{in}, u_{kl}^{end}, θ_{kl}^{t}). Therefore, the value function of the dynamic optimization problem, η, can be approximated by the following Benders cuts

$$\begin{aligned}
\eta \geq\ & a_u \sum_{klfc} \frac{\bar{t}_{fckl}^{d,p}}{N_{fe}} \Lambda_{c,N_{cp}} (\bar{u}_{fckl}^{p} - \bar{\hat{u}}_{kl}^{end,p})^2 \\
& - \sum_{k,l} (\lambda_{kl}^{1,p} (x_{kl}^{in} - \bar{\hat{x}}_{kl}^{in,p}) + \lambda_{kl}^{2,p} (x_{kl}^{end} - \bar{\hat{x}}_{kl}^{end,p}) \\
& \quad + \lambda_{kl}^{3,p} (u_{kl}^{in} - \bar{\hat{u}}_{kl}^{in,p}) \\
& + \lambda_{kl}^{4,p} (u_{kl}^{end} - \bar{\hat{u}}_{kl}^{end,p}) + \lambda_{5,p}^{kl} (\theta_{kl}^{t} - \bar{\hat{\theta}}_{kl}^{t,p})) \quad \forall\, p \in P
\end{aligned} \tag{45}$$

where p is the iteration number. The detailed derivation of the Benders cut can be found in the Supporting Information. The master problem is

$$\begin{aligned}
&\text{minimize} \quad \sum_{ikl} c_{il}^{p} \frac{W_{ikl}}{H_l} + \sum_{ijkl} c_{ij}^{trans} \theta_{kl}^{t} \frac{Z_{ijkl}}{H_l} + \sum_{ikl} c_{il}^{stor} W_{ikl} + \eta \\
&\text{subject to} \quad \text{Equations 29, 30, 31, 32, 40, 45}
\end{aligned} \tag{46}$$

Algorithm 1 Nested Generalized Benders Decomposition

```
Require: Optimization problem
Set $UB^{in} = \infty, LB^{in} = -\infty, UB^{out} = \infty, LB^{out} = -\infty$
Set tolerance and optimality gap (tol)
while $(UB^{out} - LB^{out})/LB^{out} \geq 0.01$ tol do
    while $(UB^{in} - LB^{in})/LB^{in} \geq 0.01$ tol do
        Solve problem MM (Eq. 52) and obtain $LB^{in}$
        Solve problem MS (Eq. 50) and obtain $UB^{in}$
        Add benders cut in the MM problem (Eq. 51)
    end while
    Set $LB^{out} = UB^{in}$
    Solve dynamic optimization problems for each slot and line (Eq. 44), obtain $UB^{out}$
    Add Benders cut to problem MS (Eq. 45)
    Remove Benders cuts from problem MM (Eq. 51)
end while
return Upper, lower bound and variable values
```

To guarantee that the dynamic optimization problems are feasible, we add the following constraints in the master problem

$$\theta_{kl}^{t} \geq \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \theta_{ij}^{min} z_{ijkl} \quad \forall\ k,\ l \tag{47}$$

where θ_{ij}^{min} is the minimum transition time between products i and j. We also add the following symmetry-breaking constraint, which reduces the computational time without affecting the solution[37]

$$\sum_{i=1}^{N_p} iY_{i1l} \leq \sum_{i=1}^{N_p} iy_{ikl} \quad \forall\ k \in \mathcal{I}_s,\ k > 1,\ l \in \mathcal{I}_l \tag{48}$$

Finally, we can add operational constraints to the subproblem. In this work, we add the following constraints, which constrain the change of the manipulated variables

$$\begin{aligned}
&u_{mfckl} - u_{mf-1,ckl} \leq U_m^{max}(t_{fckl}^{d} - t_{f-1,ckl}^{d}) \ \ \forall\ m, f \geq 1,\ c,\ k,\ l\\
&u_{mfckl} - u_{mfc-1,kl} \leq U_m^{max}(t_{fckl}^{d} - t_{fc-1,kl}^{d}) \ \ \forall\ m, f,\ c \geq 1,\ k,\ l\\
&u_{mfckl} - u_{mf-1,ckl} \geq U_m^{min}(t_{fckl}^{d} - t_{f-1,ckl}^{d}) \ \ \forall\ m, f \geq 1,\ c,\ k,\ l\\
&u_{mfckl} - u_{mfc-1,kl} \geq U_m^{min}(t_{fckl}^{d} - t_{fc-1,kl}^{d}) \ \ \forall\ m, f,\ c \geq 1,\ k,\ l
\end{aligned} \tag{49}$$

where $U_m^{max}$and U_m^{min} are the maximum and minimum rate of change of manipulated variable m. The master problem is a mixed integer nonlinear problem (MINLP), which is solved with BARON,[38] and the subproblems are nonlinear problems (NLPs), which are solved with IPOPT[39] in Pyomo.[40] The optimality gap tolerance is set equal to 0.1%. Finally, we note that this approach cannot guarantee global optimality, since the subproblem is nonconvex.[41]

Application of Nested Generalized Benders Decomposition Based on the Structure of the Variable Graph. Based on the structure of the ω_0 matrix, nested Generalized Benders decomposition can be applied. The original problem is first decomposed into a master and subproblem, as described in the previous section. Then, the structure of the core is used to solve the master problem using GBD. The master problem is decomposed into two subproblems that we will define as MM and MS, where MM stands for Master_Master and MS for Master_Subproblem. In this approach, the MM problem contains variables $s_4 = \{Y_{ikl},\ z_{ijkl},\ x_{kl}^{end},\ x_{kl}^{in},\ u_{kl}^{end},\ u_{kl}^{in}\}$ (complicating variables), and the other scheduling variables and associated constraints are assigned in the MS problem. Therefore, problem MS is solved for fixed values of the complicating variables s_4 and is equal to

$$\begin{aligned}
&\text{minimize} \quad \sum_{ikl} c_{il}^{p} \frac{W_{ikl}}{H_l} + \sum_{ijkl} c_{ij}^{trans} \theta_{kl}^{t} \frac{z_{ijkl}}{H_l} + \sum_{ikl} c_{il}^{stor} W_{ikl} + \eta\\
&\text{subject to}\\
&\text{Equation } 31,\ \ 32,\ 47,\ 45\\
&Y_{ikl} = \overline{Y}_{ikl} \ \ \forall\ \ i \in \mathcal{I}_p,\ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{ikl}^{1}\\
&z_{ijkl} = \overline{z}_{ijkl} \ \ \forall\ \ i \in \mathcal{I}_p,\ j \in \mathcal{I}_p,\ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{ijkl}^{2}\\
&x_{kl}^{in} = \overline{x}_{kl}^{in} \ \ \forall\ \ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{kl}^{3}\\
&x_{kl}^{end} = \overline{x}_{kl}^{end} \ \ \forall\ \ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{kl}^{4}\\
&u_{kl}^{in} = \overline{u}_{kl}^{in} \ \ \forall\ \ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{kl}^{5}\\
&u_{kl}^{end} = \overline{u}_{kl}^{end} \ \ \forall\ \ k \in \mathcal{I}_s,\ l \in \mathcal{I}_l \ :\ \ \mu_{kl}^{6}
\end{aligned} \tag{50}$$

where μ are the Lagrangean multipliers. The solution of this problem depends on the values of the complicating variables, and the value function of this problem, η_2, can be approximated by the following Benders cut

$$\begin{aligned}
&\eta_2 \geq \sum_{ikl} c_{il}^{p} \frac{\overline{W}_{ikl}^{q}}{\overline{H}_l^{q}} + \sum_{ijkl} c_{ij}^{trans} \overline{\theta}_{kl}^{t,q} \frac{\overline{z}_{ijkl}^{q}}{\overline{H}_l^{q}} + \sum_{ikl} c_{il}^{stor} \overline{W}_{ikl}^{q} + \overline{\eta}^{q}\\
&- \sum_{ikl} \mu_{ikl}^{1,q}(y_{ikl} - \overline{Y}_{ikl}^{q}) - \sum_{ijkl} \mu_{ijkl}^{2,q}(z_{ijkl} - \overline{z}q_{ijkl})\\
&- \sum_{kl} (\mu_{kl}^{3,q}(x_{kl}^{in} - \overline{x}_{ikl}^{in,q}) - \mu_{kl}^{4,q}(x_{kl}^{end} - \overline{x}_{ikl}^{end,q})\\
&- \mu_{kl}^{5,q}(u_{kl}^{in} - \overline{u}_{ikl}^{in,q}) - \mu_{kl}^{6,q}(u_{kl}^{end} - \overline{u}_{ikl}^{end,q})) \ \ \forall\ q \in Q
\end{aligned} \tag{51}$$

where q is the inner iteration number. The exact derivation of this equation can be found in the Supporting Information. The MM problem is equal to

$$\begin{aligned}
&\text{minimize } \eta_2\\
&\text{subject to eqs } 29,\ \ 30,\ \ 40,\ \ 48,\ \ 51
\end{aligned} \tag{52}$$

Problem MM is a mixed integer linear problem (MILP) solved with Gurobi,[42] and problem MS is a NLP solved with

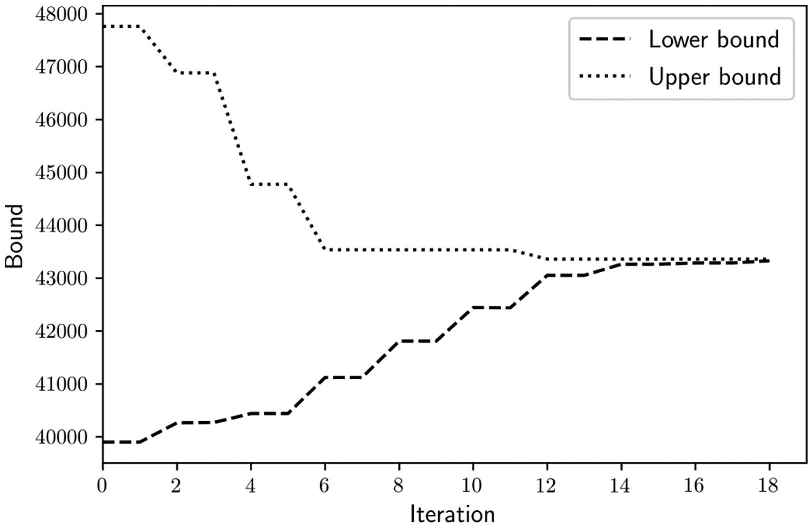


Figure 4. Evolution of the upper and lower bound for the single GBD algorithm.

IPOPT.[39] The overall nested GBD algorithm is presented in Algorithm 1. In this nested GBD approach, the number of constraints in problem MS at every iteration increases due to the addition of the Benders cuts that approximate the value function of the dynamic optimization problem. Also, once the master problem is solved, the Benders cuts (eq 51) are removed, since they may not be valid for the problem in the next iteration. The optimality gap tolerance for both the inner and outer loops is set equal to 0.1%. Similar to the case of a single GBD, the nested GBD cannot guarantee global optimality.[41]

Results. First, we solve the problem using GBD, and the results are presented in Figure 4. The algorithm converges after 209 CPU seconds (19 iterations), and the value of the objective function is 43.36 10^3 \$/h. Briefly, 99% of the CPU time is used for the solution of the master problem. The production results are presented in Table 2, and the concentration and flow rate profiles are presented in Figures 5 and 6. The cycle time in the first line is 26.89 h and that in the second line is 44.47 h. The transition times in the first line are smaller compared to the second line. Furthermore, in both lines, the majority of the production time is dedicated to the production of one product (product 1 in line 1 and product 2 in line 2). This is due to the lower production rate of these products.

The nested algorithm converges after 54 CPU seconds. The value of the objective function is 43.36 10^3 \$/h, and the evolution of the upper and lower bounds is presented in Figure 7. The production results are the same as the ones obtained with the single GBD algorithm (Table 2). Finally, we compare the evolution of the upper and lower bounds with CPU time for the GBD and nested GBD approaches. From Figure 8, we see that the exploitation of the hybrid multicore community structure of the problem reduces the computational time by 74%. Finally, we note that solving the monolithic problem with BARON after 3000 CPU seconds, the gap is 88%, and the value of the objective function is 74.4 10^3 \$/h. These results highlight the importance of detecting and exploiting the true underlying structure of an optimization problem.

Table 2. Production Results

slot	product	production amount (kg)	production time (h)	transition time (h)
		Line 1, Cycle Time 26.89 h		
1	1	912	24.3	1.50
2	3	457	2.58	0.93
		Line 2, Cycle Time 44.47 h		
1	4	711	3.57	2.15
2	2	2341	40.90	4.87

INTEGRATION OF PLANNING, SCHEDULING, AND DYNAMIC OPTIMIZATION

Problem Formulation. We now consider the problem of integration of planning, scheduling, and dynamic optimization. The detailed explanation of the model can be found in Gutiérrez-Limón et al.[43] We assume that the number of products is N_p ($i = \{1,..,N_p\}$), the number of planning periods is N_{pr} (p = $\{1,..,N_{pr}\}$), and the number of slots is N_s ($k = \{1,..,N_s\}$). First, we define the binary variable W_{ikp}, which is equal to 1 if product i is produced at slot k in period p and zero, otherwise. We also define the binary variable Z_{ijkp}, which is equal to 1 if a product i is followed by product j in slot k in period p and the binary variable Zp_{ijp}, which is equal to one if transition occurs between product i and j between time periods. At each time slot, only one product can be produced, which is enforced with the following constraints

$$\sum_i W_{ikp} = 1 \quad \forall\ k,\ p \tag{53}$$

The transitions between the products are modeled through the following equations

$$\begin{gathered} Z_{ijkp} \geq W_{ikp} + W_{j,k+1,p} - 1 \quad \forall\ i,\ j \in N_p,\ i \neq j,\ k \neq N_s,\ p \\ Zp_{ijkp} \geq W_{iN_sp} + W_{j,1,p+1} - 1 \quad \forall\ i,\ j \in N_p,\ i \neq j,\ p \neq N_{per} \end{gathered} \tag{54}$$

The production time of product i in slot k in period p is θ_{ikp}, and the production time of product i in period p is $\hat{\theta}_{ip}$. The starting, ending, and transition times in slot k in period p *are* T^s_{kp}, $T^e_{k,p}$, and θ^t_{kp}, respectively. The timing constraints are the following

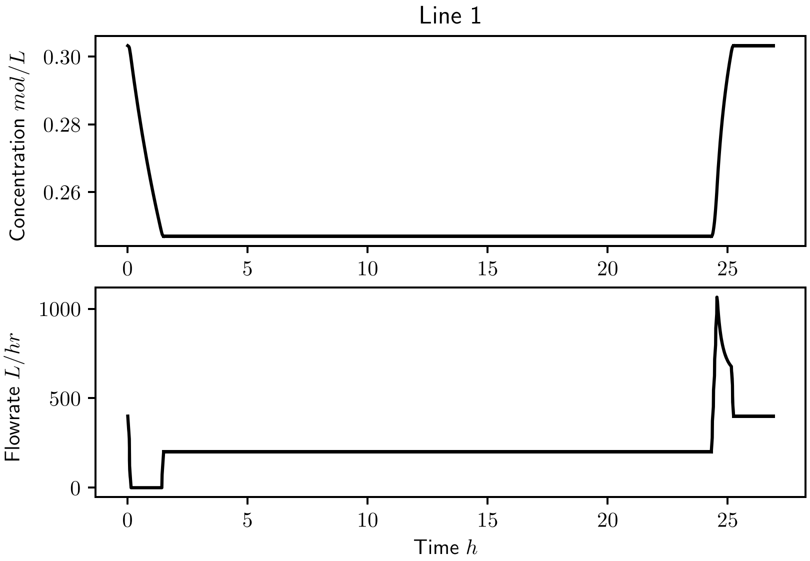


Figure 5. Concentration and inlet flow rate profile in line 1.

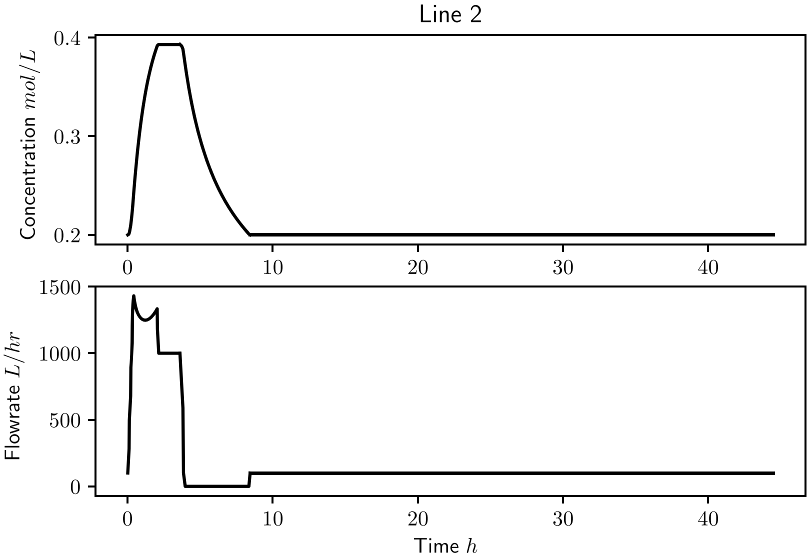


Figure 6. Concentration and inlet flow rate profile in line 2.

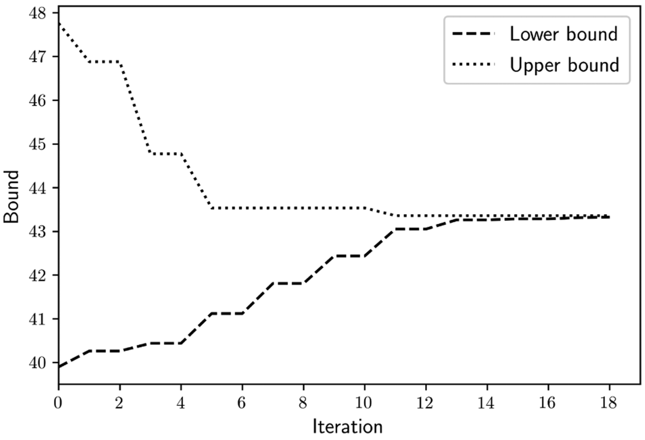


Figure 7. Evolution of the upper and lower bound for the nested GBD.

$$
\begin{aligned}
&T^{s}_{1,1} = 0\\
&T^{e}_{k,p} = T^{s}_{k,p} + \sum_{i} \theta_{ikp} + \theta^{t}_{kp} \quad \forall\ k,\ p\\
&T^{s}_{k+1,p} = T^{e}_{k,p} \quad \forall\ k \neq N_s,\ p\\
&T^{s}_{1,p+1} = T^{e}_{N_s,p} \quad \forall\ k,\ p \neq N_{per}\\
&\hat{\theta}_{ip} = \sum_{k} \theta_{ikp} \quad \forall\ i,\ p\\
&T^{e}_{k,p} \leq H_p
\end{aligned}
\tag{55}
$$

where H_p is the duration of the planning horizon. The production rate of product i is r_i, the amount of product i produced in slot k at period p is $\hat{q}_{ikp}$, and the amount of product i produced in period p is q_{ip}. The production and inventory constraints are

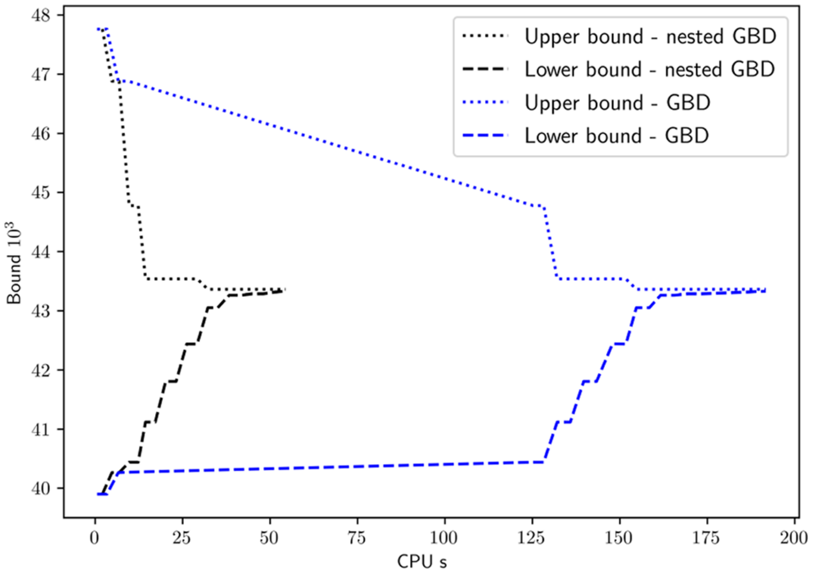


Figure 8. Evolution of the upper and lower bound for the single and nested decomposition algorithm.

$$
\begin{aligned}
&\hat{q}_{ikp} = r_i \hat{\theta}_{ikp} \quad \forall\ i,\ k,\ p\\
&q_{ip} = \sum_k \hat{q}_{ikp} \quad \forall\ i,\ p\\
&I_{ip} = I_{ip-1} + q_{ip} - S_{ip} \quad \forall\ i,\ p\\
&A_{ip} = H_p(I_{ip-1} - S_{ip-1}) + q_{ip}H_p \quad \forall\ i,\ p
\end{aligned}
\tag{56}
$$

where I_{ip} is the inventory of product i in period p, A_{ip} is the linear overestimation of the integral of inventory, and S_{ip} is the amount of product i sold in period p. The following symmetry-breaking constraints are also included

$$
\begin{aligned}
&Y_{ip} \geq W_{ikp} \quad \forall\ i,\ k,\ p\\
&Y_{ip} \leq N_{ip} \quad \forall\ i,\ p\\
&N_{ip} \geq N - \left(\sum_i Y_{ip} - 1\right) - M(1 - W_{i1p}) \quad \forall\ i,\ p\\
&N_{ip} \leq N - \left(\sum_i Y_{ip} - 1\right) + M(1 - W_{i1p}) \quad \forall\ i,\ p\\
&N_{ip} = \sum_k W_{ikp} \quad \forall\ i,\ p
\end{aligned}
\tag{57}
$$

The dynamic behavior of the system is modeled as in the previous case study, the differential equations are discretized using collocation on finite elements, and the constraints are

$$
\begin{aligned}
&x_{fckp}^{n} = x0_{fkp}^{n} + h_{kp}^{fe} \sum_{m=1}^{N_{cp}} \Omega_{mc} \dot{x}_{fmkp}^{n} \quad \forall\ n,\ f,\ c,\ k,\ p\\
&x0_{fkp}^{n} = x0_{f-1kp}^{n} + h_{kp}^{fe} \sum_{m=1}^{N_{cp}} \Omega_{mc} \dot{x}_{f-1,mkp}^{n} \quad \forall\ n,\ f \geq 2,\ c,\ k,\ p\\
&\dot{x}_{fckp}^{n} = f^{n}(x_{fckp}^{n},\ u_{fckp}^{m}) \quad \forall\ n,\ f,\ c,\ k,\ p\\
&t_{fckp}^{d} = h_{kp}^{fe}(f - 1 + \gamma_c) \quad \forall\ f,\ c,\ k,\ p
\end{aligned}
\tag{58}
$$

$$
h_{kp}^{fe} = \frac{\theta_{kp}^{t}}{N_{fe}} \quad \forall\ k,\ p
\tag{59}
$$

The dynamic model is integrated with the planning/scheduling problem through the following constraints

$$
\begin{aligned}
&x_{n,k,p}^{in} = \sum_i x_i^{ss} W_{i,k,p} \quad \forall\ n,\ k,\ p\\
&x_{n,k,p}^{end} = \sum_i x_i^{ss} W_{i,k+1,p} \quad \forall\ n,\ k,\ p\\
&u_{n,k,p}^{in} = \sum_i u_i^{ss} W_{i,k,p} \quad \forall\ n,\ k,\ p\\
&u_{n,k,p}^{end} = \sum_i u_i^{ss} W_{i,k+1,p} \quad \forall\ n,\ k,\ p
\end{aligned}
\tag{60}
$$

$$
\begin{aligned}
&x0_{1kp}^{n} = x_{n,k,p}^{in} \quad \forall\ n,\ k,\ p\\
&x_{N_{fe}N_{cp}kp}^{n} = x_{n,k,p}^{end} \quad \forall\ n,\ k,\ p\\
&u_{11kp}^{m} = u_{m,k,p}^{in} \quad \forall\ m,\ k,\ p\\
&u_{N_{fe}N_{cp}kp}^{m} = u_{m,k,p}^{end} \quad \forall\ m,\ k,\ p
\end{aligned}
\tag{61}
$$

The objective is to maximize the profit, which is equal to

$$
\begin{aligned}
f = &\sum_{i=1}^{N_p} \sum_{p=1}^{N_{per}} P_{ip} S_{ip} - \sum_{i=1}^{N_p} \sum_{p=1}^{N_{per}} C_{ip}^{oper} q_{ip}\\
&-C^{inv} \sum_{i=1}^{N_p} \sum_{p=1}^{N_{per}} A_{ip} - \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \sum_{k=1}^{N_s} \sum_{p=1}^{N_{per}} C_{ij}^{trans} Z_{ijkp}\\
&- \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \sum_{p=1}^{N_{per}} C_{ij}^{trans} Zp_{ijp}\\
&-\alpha_u \sum_{p=1}^{N_{per}} \sum_{k=1}^{N_p} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} N_{fe}^{-1} t_{fckp}^{d} \Lambda_{c,N_{cp}} (u_{fckp} - u_{kp}^{end})^2
\end{aligned}
\tag{62}
$$

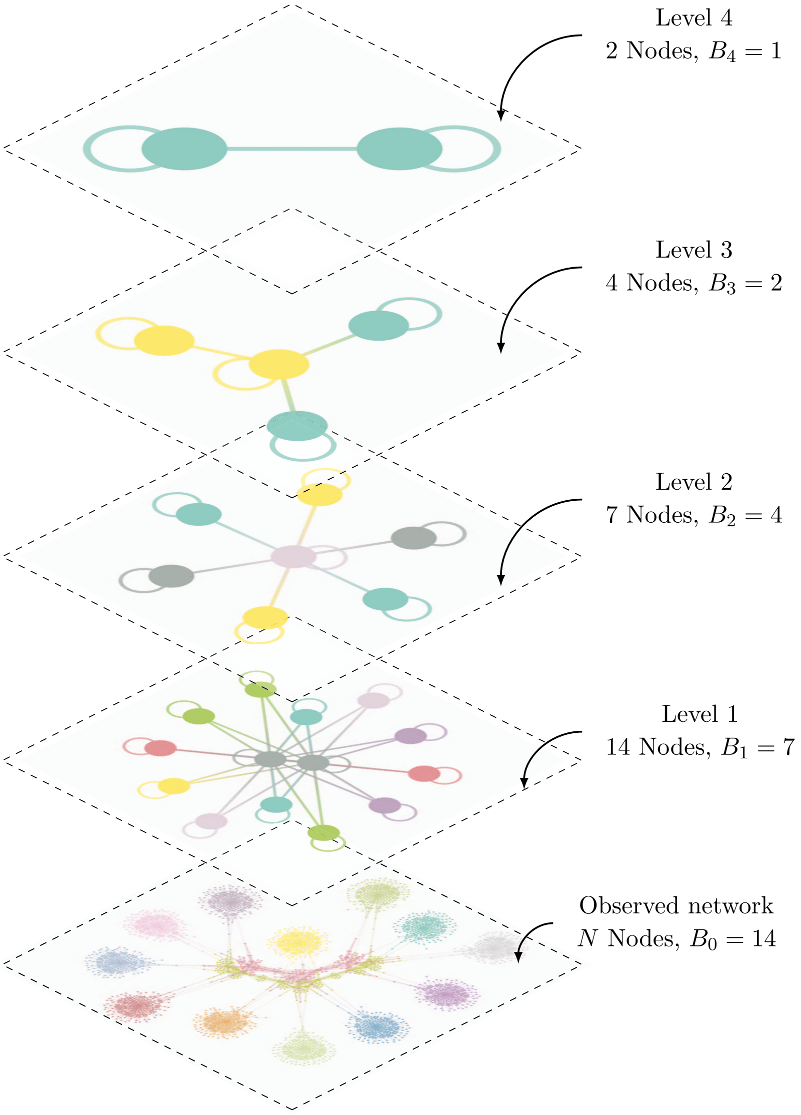


Figure 9. Inferred nSBM model of the variable graph for the integrated planning, scheduling, and dynamic optimization problem. The hollow cycles indicate self-edges.

where P_{ip} is the price of product i in period p, C_{ip}^{oper} is the operating cost of product i in period p, C^{inv} is the inventory cost, C_{ij}^{trans} is the transition cost from product i to j, and a_u is a weight coefficient. The goal of the optimization problem is to maximize eq 62 subject to constraints 53−61.

Application of Nested Stochastic Blockmodeling. The structure of the variable graph will be analyzed using degree-corrected nSBM and Bayesian inference in graph-tool.[36] We will assume that four products (4 slots) must be produced in three planning periods, and the system is the same as in the previous case study (one isothermal continuous stirred reactor). Application of nSBM identifies four levels (Figure 9). The observed network is partitioned into 14 blocks (Figure 10).

The multigraph in the first level is partitioned into seven blocks, and in the second level, into four blocks. The number of edges between the blocks in the observed and the different multigraphs is given by ω_0, ω_1, and ω_2, respectively. This partition of the variable graph into four levels provides information about the multiscale nature of the problem and the different structures that are present at different hierarchical levels.

$$\omega_0 = \begin{bmatrix} 842 & 48 & 3 & 3 & \dots & 3 \\ 48 & 742 & 1 & 1 & \dots & 1 \\ 3 & 1 & 757 & 0 & \dots & 0 \\ 3 & 1 & 0 & 757 & \dots & 0 \\ \vdots & \vdots & \vdots & & \ddots & \\ 3 & 1 & 0 & \dots & \dots & 757 \end{bmatrix} \in \mathbb{R}^{14\times 14}$$

$$\omega_1 = \begin{bmatrix} 1672 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 1514 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 1514 & 0 & 0 & 0 & 0 \\ 8 & 0 & 0 & 1514 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 1514 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 1514 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 1514 \end{bmatrix}$$

$$\omega_2 = \begin{bmatrix} 1672 & 16 & 16 & 16 \\ 16 & 3028 & 0 & 0 \\ 16 & 0 & 3028 & 0 \\ 16 & 0 & 0 & 3028 \end{bmatrix}$$

The original graph is decomposed into 14 blocks, and the planning and scheduling variables are assigned in the two middle blocks. Variables W_{ikl}, Z_{ijkl}, Zp_{ijp}, Y_{ip}, N_{ip}, x^{in}_{kp}, x^{end}_{kp}, u^{in}_{kp}, and u^{end}_{kp} are assigned in the green block, and the other planning/scheduling variables, in the other blocks. The variables associated with the dynamic behavior of the problem for each slot and period (x_{nfckp}, u_{mfckp}, h^{fe}_{kp}, t_{fckp}) are assigned in the blocks in the periphery. From the structure of the ω_0 matrix, we can determine that the graph has a hybrid multicore community structure. The variables for the dynamic optimization problems for each slot and period are assigned to different blocks. The variables in these blocks are densely coupled, denoting a community structure. However, these blocks are weakly coupled with the planning/scheduling variables, which form a multicore structure, since these variables are assigned into two blocks and are connected with all of the dynamic optimization variables.

In the first level, the planning and scheduling variables are assigned into the same block, and the dynamic optimization variables are assigned into different blocks, leading to a core−periphery structure, highlighted by the L shape of the ω_1 matrix. Similarly, the multigraph in the second level has a core−periphery structure. In this multigraph, the planning/scheduling variables are in the middle node, and the variables for the dynamic optimization problems are in the periphery. This graph is a coarser partition of the level 1 graph, and although each node in the periphery corresponds to the variables of two dynamic optimization problems, these variables are decoupled, since, in the first level, the nodes in the periphery are not coupled directly, i.e., there does not exist an edge between the nodes in the periphery.

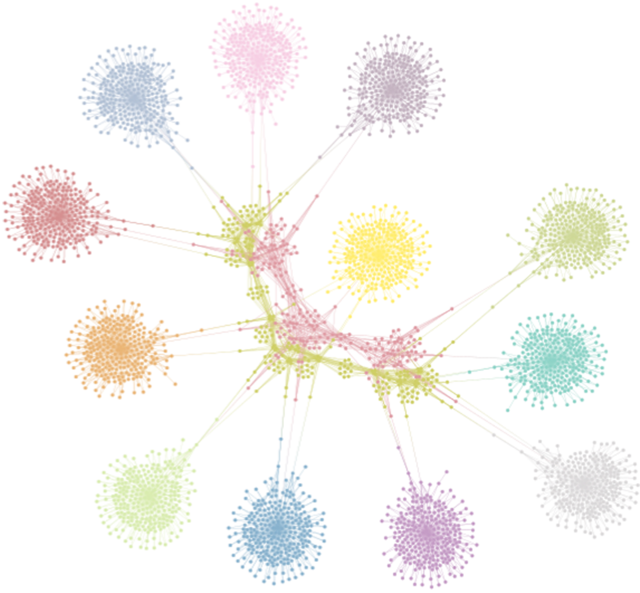


Figure 10. Partition of the variable graph for the integrated planning, scheduling, and dynamic optimization problem.

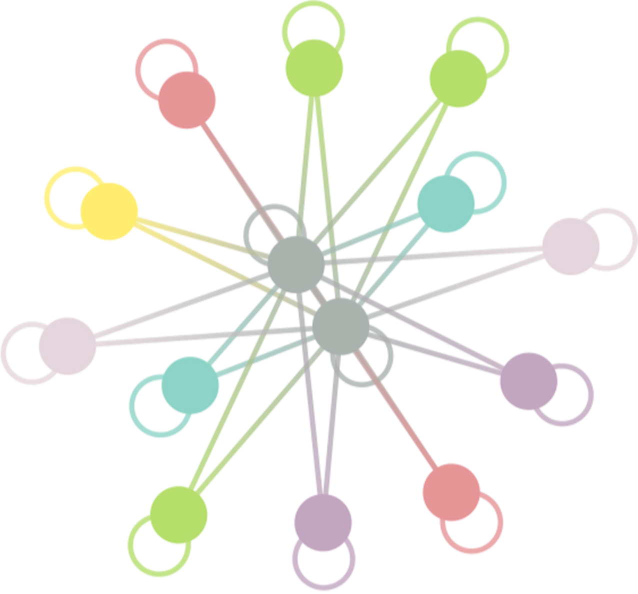


Figure 11. Partition of the first level multigraph for the integrated planning, scheduling, and dynamic optimization problem.

Application of Generalized Benders Decomposition Based on the Core−Periphery Structure of the First Level Multigraph. The core−periphery structure of the multigraph in level 1 can be used as the basis for the application of GBD. The variables in the core (gray nodes in Figure 11) correspond to the planning/scheduling variables and are assigned in the master problem. The variables for the dynamic optimization problems for each slot and period, and

Table 3. Operating Conditions and Product Price for the Integrated Planning, Scheduling, and Dynamic Optimization Problem

product	c^{ss} (mol/L)	Q^{ss} (L/h)	production rate
A	0.24	200	150
B	0.2	100	80
C	0.30	400	278
D	0.393	1000	607

Table 4. Operating and Transition Cost for the Integrated Planning, Scheduling, and Dynamic Optimization Problem, C^{inv} = 0.026, a_u = 1

	C^{oper}			C^{trans}			
product	p = 1	p = 2	p = 3	A	B	C	D
A	13	13	13	0	100	60	120
B	22	12	12	150	0	50	80
C	35	45	45	200	150	0	100
D	29	19	19	90	100	120	0

Table 5. Product Demand for the Integrated Planning, Scheduling, and Dynamic Optimization Problem

	demand (mol/week)			price ($/mol)		
prod.	p = 1	p = 2	p = 3	p = 1	p = 2	p = 3
A	6000	8000	7000	200	220	200
B	5000	3600	6000	160	140	150
C	7000	9000	7000	130	150	140
D	4000	11 000	11 000	110	110	120

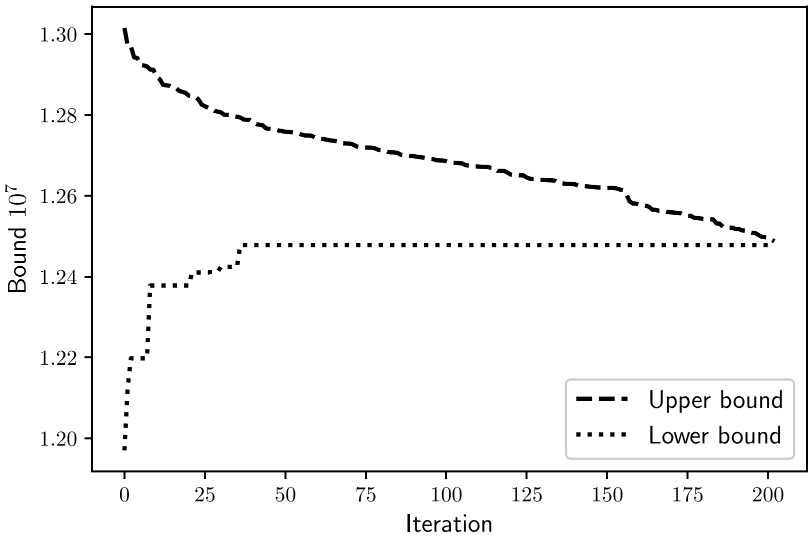


Figure 12. Convergence of Generalized Benders decomposition based on the core−periphery structure of the first level multigraph for the integrated planning, scheduling, and dynamic optimization problem.

the associated constraints, are assigned in the subproblem. The complicating variables are x_{kp}^{in}, x_{kp}^{end}, u_{kp}^{in}, u_{kp}^{end}, and θ_{kp}^{t}. The subproblem is solved for fixed values of the complicating variables and is

$$\text{minimize} \;\; \alpha_u \sum_{p=1}^{N_{per}} \sum_{k=1}^{N_s} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} N_{fe}^{-1} t_{fckp}^{d} \Omega_{c,N_{cp}} (u_{fckp} - u_{kp}^{end})^2$$

$$\text{subject to}$$

$$\text{Equations 58, 59, 61}$$

$$x_{nkp}^{in} = \overline{x}_{nkp}^{in} \quad \forall\ n,\ k,\ p\ :\ \gamma_{kl}^{1}$$

$$x_{nkp}^{end} = \overline{x}_{nkp}^{end} \quad \forall\ n,\ k,\ p\ :\ \gamma_{kl}^{2}$$

$$u_{mkp}^{in} = \overline{u}_{mkp}^{in} \quad \forall\ m,\ k,\ p\ :\ \gamma_{kl}^{3}$$

$$u_{mkp}^{end} = \overline{u}_{mkp}^{end} \quad \forall\ m,\ k,\ p\ :\ \gamma_{kl}^{4}$$

$$\theta_{kp}^{t} = \overline{\theta}_{kp}^{t} \quad \forall\ k,\ p\ :\ \gamma_{kl}^{5} \tag{63}$$

where γ is the Lagrangean multiplier of each constraint. This problem can be solved independently for every slot and period. The value function of this problem is approximated by the following Benders cut

$$\eta \geq \alpha_u \sum_{p=1}^{N_{per}} \sum_{k=1}^{N_p} \sum_{f=1}^{N_{fe}} \sum_{c=1}^{N_{cp}} N_{fe}^{-1} \overline{t}_{fckp}^{d,\nu} \Omega_{c,N_{cp}} (\overline{u}_{fckp}^{\nu} - \overline{u}_{kp}^{end,,\nu})^2$$
$$- \sum_{p=1}^{N_{per}} \sum_{k=1}^{N_s} (\gamma_{k,p}^{1,\nu} (x_{kp}^{in} - \overline{x}_{kp}^{in,\nu}) + \gamma_{k,p}^{2,\nu} (x_{kp}^{end} - \overline{x}_{kp}^{end,\nu})$$
$$+ \gamma_{k,p}^{3,\nu} (u_{kp}^{in} - \overline{u}_{kp}^{in,\nu}) + \gamma_{k,p}^{4,\nu} (u_{kp}^{end} - \overline{u}_{kp}^{end,\nu})$$
$$+ \gamma_{k,p}^{5,\nu} (\theta_{kp}^{t} - \overline{\theta}_{kp}^{t,\nu})) \quad \forall\ \nu \in \mathcal{V} \tag{64}$$

where the superscript ν is the iteration number. The master problem is

$$\text{maximize} \;\; \sum_{i=1}^{N_p} \sum_{p=1}^{N_{per}} (P_{ip} S_{ip} - C_{ip}^{oper} q_{ip} - C^{inv} A_{ip})$$
$$- \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \sum_{k=1}^{N_s} \sum_{p=1}^{N_{per}} C_{ij}^{trans} Z_{ijkp}$$
$$- \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} \sum_{p=1}^{N_{per}} C_{ij}^{trans} Zp_{ijp} - \eta$$
$$\text{subject to Eq. 53, 54, 55, 56, 57, 60, 64} \tag{65}$$

To guarantee that the dynamic optimization problem is feasible, we add the following constraint in the master problem

$$\theta_{kp}^{t} \geq \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} Z_{ijkp} \theta_{ij}^{min} \quad \forall\ k,\ p,\ k \neq N_s$$

$$\theta_{N_s,p}^{t} \geq \sum_{i=1}^{N_p} \sum_{j=1}^{N_p} Zp_{ijp} \theta_{ij}^{min} \quad \forall\ p,\ p \neq N_{per} \tag{66}$$

Finally, we add operational constraints similar to eq 49. The exact derivation of the master, subproblem, and Benders cut can be found in the Supporting Information. The master problem is a MILP solved with Gurobi,[4242] and the subproblem is an NLP solved with IPOPT[39] in Pyomo.[40] The problem is solved to 0.1% optimality gap.

Results. We solve the integrated problem using the GBD approach proposed in the previous section, and the economic parameters of the optimization problem are presented in Tables 3−45. The algorithm converges after 294 CPU seconds, the evolution of the upper and lower bounds is presented in Figure 12, and the value of the objective function is 1.247 10^7\$. The solution of the master problem accounts for 76% of the total CPU time. A monolithic solution with BARON[38] cannot find a feasible solution after 500 CPU seconds.

The production results are presented in Table 6, and the profiles of the concentration and inlet flow rate for each slot

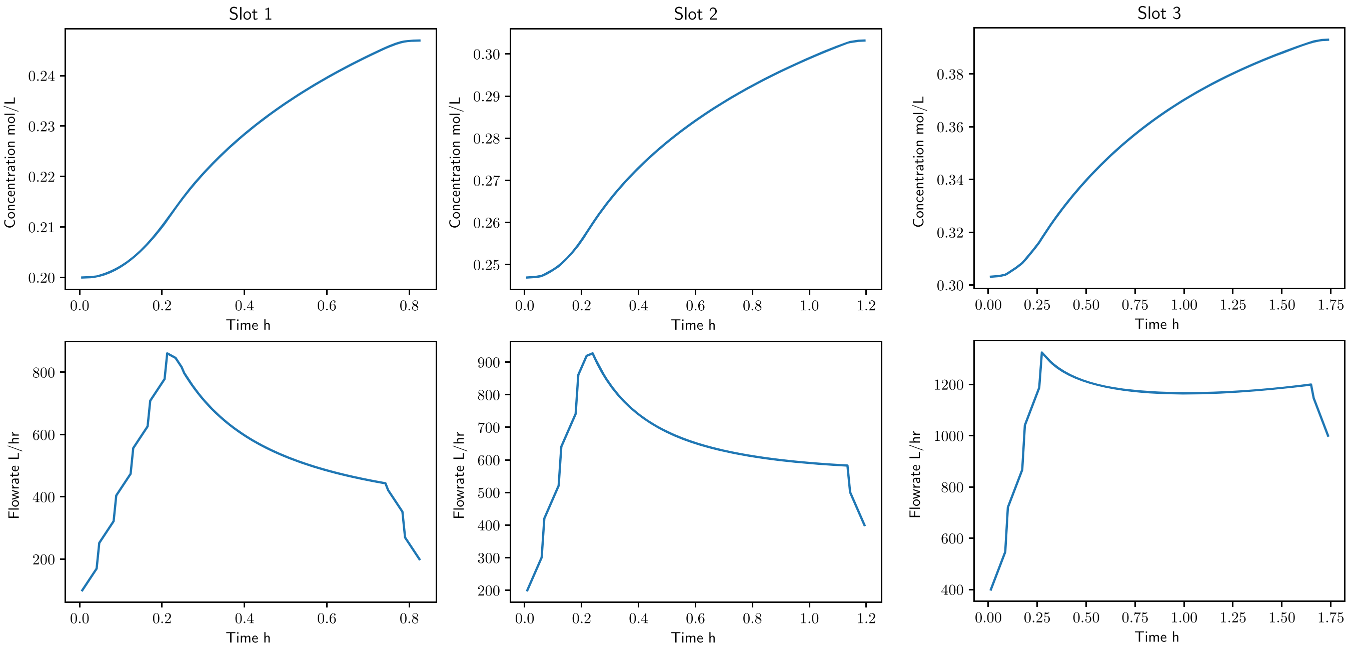


Figure 13. Concentration and inlet flow rate profile for the first period.

Table 6. Production Results for the Integrated Planning, Scheduling, and Dynamic Optimization Problem

slot	product	production amount (mol)	production time (h)	transition time (h)
		Period 1		
1	B	5000	62.50	0.82
2	A	6000	39.83	1.19
3	C	15 418	55.31	1.73
4	D	4000	6.59	0
		Period 2		
1	D	11 000	18.12	1.33
2	C	581.85	2.08	1.50
3	A	8000	53.11	2.24
4	B	5000	62.5	0
		Period 3		
1	B	4600	57.5	1.08
2	A	8000	53.11	1.16
3	C	9000	32.29	2.01
4	D	29 089	47.92	0

and period are presented in Figures 13, 14, and 15. No transition occurs between the different time periods, leading to a reduction in the transition cost. The production of the products satisfies the demand. Furthermore, we note the production profiles of products *C* and *D*. Product *C* is overproduced in the first period, since its operating cost is lower compared to the other two periods. Product *D* is overproduced in the last period, where its operating cost is lower, and the price is higher compared to the other periods.

From these results, we can argue that application of GBD based on the core−periphery structure of level 1 of the variable graph enables an efficient solution of the integrated problem.

Remark 1. We note that in this case study, similar to the one in the integration of scheduling and dynamic optimization, we can apply a nested GBD approach based on the hybrid multicore community structure of the variable graph. We did not apply nested GBD, since the master problem is a MILP, which can be solved efficiently with Gurobi. For problems with a larger number of products/periods, application of nested GBD might be necessary to further reduce the computational time.

CONCLUSIONS AND FURTHER REMARKS

The integration of process operations leads to large-scale optimization problems whose monolithic solution is challenging. In this work, we proposed nested stochastic blockmodeling and Bayesian inference as a framework to detect the underlying hierarchical block structure and the hierarchy itself of such optimization problems. We applied this framework to representative problems on integration of scheduling and dynamic optimization and planning, scheduling, and dynamic optimization. The inference and solution results highlight the inherent ability of the proposed approach to detect the multiscale nature of these problems and the complex block structures that are present in the different hierarchical levels. Furthermore, we showed that the exploitation of the structure at different hierarchical levels enables an efficient solution of such problems using decomposition-based solution algorithms. Finally, the following general remarks can be made.

Remark 2. The decomposition that is obtained with this approach is supported by statistical evidence, and when the decomposition with the minimum description length is selected, it is optimal from a network structure perspective. Despite the improvement in the computational time noted compared to the monolithic solution of the problem, it is not guaranteed that the obtained decomposition is optimal with respect to the computational time or convergence rate. This problem will be addressed in future work.

Remark 3. For the integrated scheduling and dynamic optimization problem, we applied weighted nested SBM using a discrete geometric model for the edge weights. In general, different models can be selected, such as discrete binomial and Poisson and real exponential and normal.[34] Since the

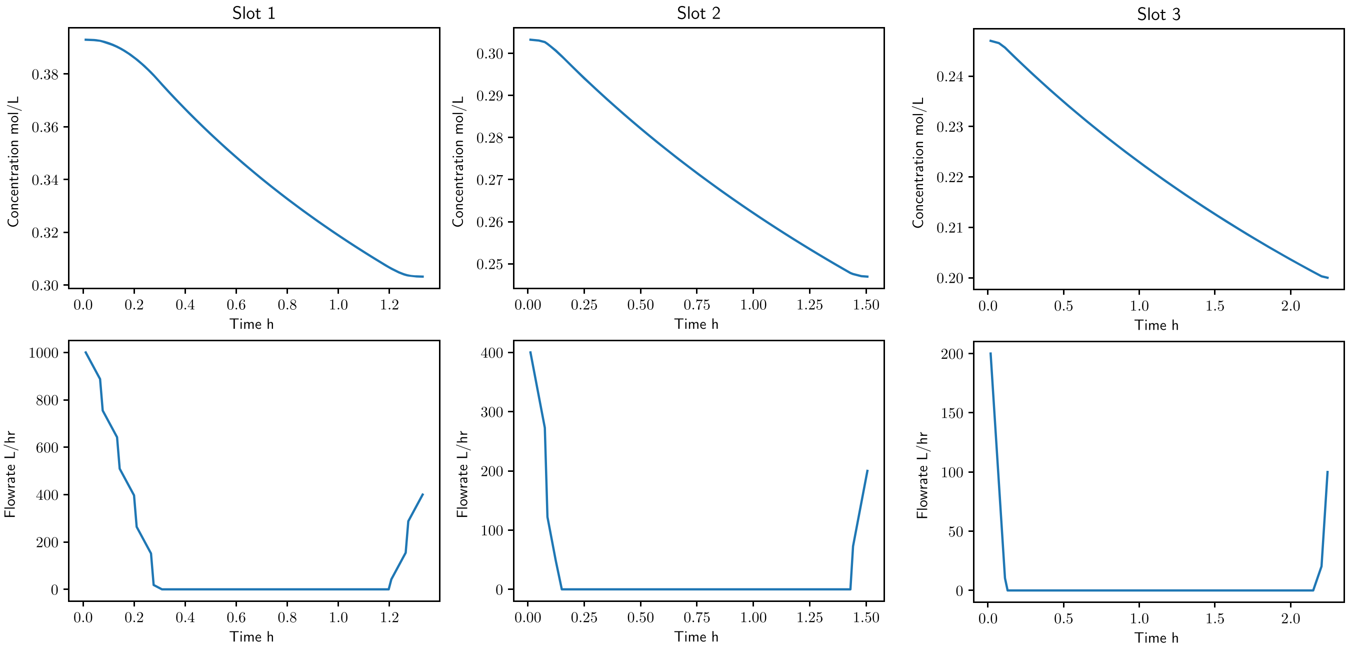


Figure 14. Concentration and inlet flow rate profile for the second period.

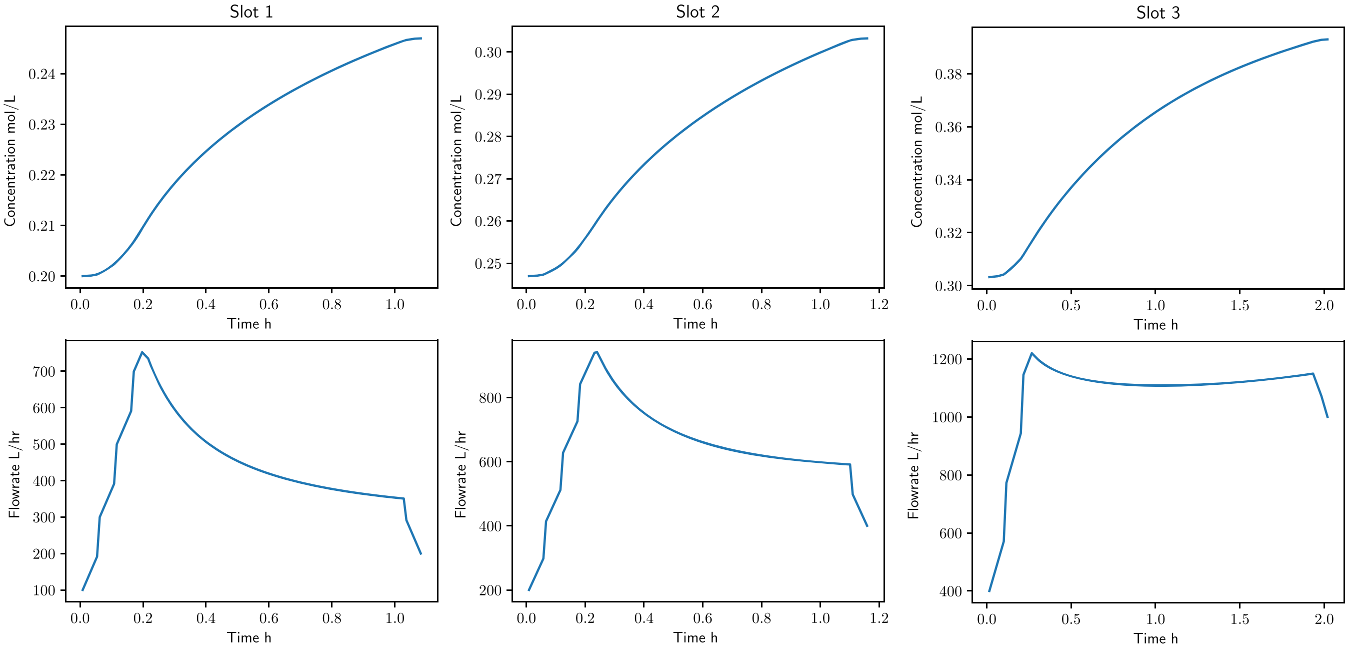


Figure 15. Concentration and inlet flow rate profile for the third period.

selection of the model for the weights is an assumption, one can try different models and test the suitability of the resulting decompositions for the solution of the problem.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.iecr.1c01570.

Formulation of the master and subproblem for the application of Generalized Benders decomposition (GBD) and nested GBD for the two case studies (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Prodromos Daoutidis − *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States;* orcid.org/0000-0003-4803-0404; Email: daout001@umn.edu

Author

Ilias Mitrai − *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.1c01570

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

Financial support from NSF-CBET (Award Number 1926303) is gratefully acknowledged.

REFERENCES

(1) Grossmann, I. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE J.* **2005**, *51*, 1846−1857.

(2) Daoutidis, P.; Lee, J. H.; Harjunkoski, I.; Skogestad, S.; Baldea, M.; Georgakis, C. Integrating operations and control: A perspective and roadmap for future research. *Comput. Chem. Eng.* **2018**, *115*, 179−184.

(3) Maravelias, C. T.; Sung, C. Integration of production planning and scheduling: Overview, challenges and opportunities. *Comput. Chem. Eng.* **2009**, *33*, 1919−1930.

(4) Bhosekar, A.; Ierapetritou, M. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Comput. Chem. Eng.* **2018**, *108*, 250−267.

(5) Pattison, R. C.; Touretzky, C. R.; Johansson, T.; Harjunkoski, I.; Baldea, M. Optimal process operations in fast-changing electricity markets: framework for scheduling with low-order dynamic models and an air separation application. *Ind. Eng. Chem. Res.* **2016**, *55*, 4562−4584.

(6) Zhuge, J.; Ierapetritou, M. G. Integration of scheduling and control for batch processes using multi-parametric model predictive control. *AIChE J.* **2014**, *60*, 3169−3183.

(7) Burnak, B.; Katz, J.; Diangelakis, N. A.; Pistikopoulos, E. N. Simultaneous process scheduling and control: a multiparametric programming-based approach. *Ind. Eng. Chem. Res.* **2018**, *57*, 3963−3976.

(8) Chu, Y.; You, F. Integrated planning, scheduling, and dynamic optimization for batch processes: MINLP model formulation and efficient solution methods via surrogate modeling. *Ind. Eng. Chem. Res.* **2014**, *53*, 13391−13411.

(9) Charitopoulos, V. M.; Dua, V.; Papageorgiou, L. G. Traveling salesman problem-based integration of planning, scheduling, and optimal control for continuous processes. *Ind. Eng. Chem. Res.* **2017**, *56*, 11186−11205.

(10) Allman, A.; Palys, M. J.; Daoutidis, P. Scheduling-informed optimal design of systems with time-varying operation: A wind-powered ammonia case study. *AIChE J.* **2019**, *65*, No. e16434.

(11) Guignard, M.; Kim, S. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Math. Program.* **1987**, *39*, 215−228.

(12) van den Heever, S. A.; Grossmann, I. E.; Vasantharajan, S.; Edwards, K. A Lagrangean decomposition heuristic for the design and planning of offshore hydrocarbon field infrastructures with complex economic objectives. *Ind. Eng. Chem. Res.* **2001**, *40*, 2857−2875.

(13) Gupta, A.; Maranas, C. D. A hierarchical Lagrangean relaxation procedure for solving midterm planning problems. *Ind. Eng. Chem. Res.* **1999**, *38*, 1937−1947.

(14) Li, Z.; Ierapetritou, M. G. Production planning and scheduling integration through augmented Lagrangian optimization. *Comput. Chem. Eng.* **2010**, *34*, 996−1006.

(15) Benders, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* **1962**, *4*, 238−252.

(16) Geoffrion, A. M. Generalized benders decomposition. *J. Optim. Theory Appl.* **1972**, *10*, 237−260.

(17) Chu, Y.; You, F. Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming. *Comput. Chem. Eng.* **2013**, *58*, 315−333.

(18) Li, X.; Chen, Y.; Barton, P. I. Nonconvex generalized benders decomposition with piecewise convex relaxations for global optimization of integrated process design and operation problems. *Ind. Eng. Chem. Res.* **2012**, *51*, 7287−7299.

(19) Iyer, R. R.; Grossmann, I. E. A bilevel decomposition algorithm for long-range planning of process networks. *Ind. Eng. Chem. Res.* **1998**, *37*, 474−481.

(20) Shi, H.; Chu, Y.; You, F. Novel optimization model and efficient solution method for integrating dynamic optimization with process operations of continuous manufacturing processes. *Ind. Eng. Chem. Res.* **2015**, *54*, 2167−2187.

(21) Conejo, A. J.; Castillo, E.; Minguez, R.; Garcia-Bertrand, R. *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*; Springer, 2006.

(22) Daoutidis, P.; Tang, W.; Allman, A. Decomposition of control and optimization problems by network structure: concepts, methods and inspirations from biology. *AIChE J.* **2019**, *65*, No. e16708.

(23) Allman, A.; Tang, W.; Daoutidis, P. DeCODe: a community-based algorithm for generating high-quality decompositions of optimization problems. *Optim. Eng.* **2019**, *20*, 1067−1084.

(24) Mitrai, I.; Tang, W.; Daoutidis, P. Stochastic Blockmodeling for Learning the Structure of Optimization Problems. *AIChE J.* **2021**, No. e17415.

(25) Fortunato, S.; Hric, D. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1−44.

(26) Tang, W.; Allman, A.; Pourkargar, D. B.; Daoutidis, P. Optimal decomposition for distributed optimization in nonlinear model predictive control through community detection. *Comput. Chem. Eng.* **2018**, *111*, 43−54.

(27) Mitrai, I.; Daoutidis, P. Decomposition of integrated scheduling and dynamic optimization problems using community detection. *J. Process Control* **2020**, *90*, 63−74.

(28) Goldenberg, A.; Zheng, A. X.; Fienberg, S. E.; Airoldi, E. M.; et al. A Survey of Statistical Network Models. *Found. Trends Mach. Learn.* **2009**, *2*, 129−233.

(29) Peixoto, T. P. Hierarchical block structures and high-resolution model selection in large networks. *Phys. Rev. X* **2014**, *4*, No. 011047.

(30) Peixoto, T. P. Bayesian Stochastic Blockmodeling. In *Advances in Network Clustering and Blockmodeling*; Wiley, 2019; pp 289−332.

(31) Holland, P. W.; Laskey, K. B.; Leinhardt, S. Stochastic blockmodels: First steps. *Soc. Networks* **1983**, *5*, 109−137.

(32) Peixoto, T. P. Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E* **2017**, *95*, No. 012317.

(33) Peixoto, T. P. Entropy of stochastic blockmodel ensembles. *Phys. Rev. E* **2012**, *85*, No. 056122.

(34) Peixoto, T. P. Nonparametric weighted stochastic block models. *Phys. Rev. E* **2018**, *97*, No. 012306.

(35) Flores-Tlacuahuac, A.; Grossmann, I. E. Simultaneous scheduling and control of multiproduct continuous parallel lines. *Ind. Eng. Chem. Res.* **2010**, *49*, 7909−7921.

(36) Peixoto, T. P. The Graph-Tool Python Library, 2014. http://figshare.com/articles/graph_tool/1164194 (accessed April 26, 2021).

(37) Zhuge, J.; Ierapetritou, M. In *Simultaneous Scheduling and Control with Closed Loop Implementation on Parallel Units*, Proceedings Foundations of Computer-Aided Process Operations (FOCAPO), Savannah, USA, 2012.

(38) Tawarmalani, M.; Sahinidis, N. V. A polyhedral branch-and-cut approach to global optimization. *Math. Program.* **2005**, *103*, 225−249.

(39) Wächter, A.; Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25−57.

(40) Hart, W. E.; Laird, C. D.; Watson, J.-P.; Woodruff, D. L.; Hackebeil, G. A.; Nicholson, B. L.; Siirola, J. D. *Pyomo-Optimization Modeling in Python*, 2nd ed.; Springer Science & Business Media, 2017; Vol. *67*.

(41) Sahinidis, N.; Grossmann, I. E. Convergence properties of generalized Benders decomposition. *Comput. Chem. Eng.* **1991**, *15*, 481−491.

(42) Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. https://www.gurobi.com (accessed April 26, 2021).

(43) Gutiérrez-Limón, M. A.; Flores-Tlacuahuac, A.; Grossmann, I. E. MINLP formulation for simultaneous planning, scheduling, and control of short-period single-unit processing systems. *Ind. Eng. Chem. Res.* **2014**, *53*, 14679−14694.