Identification of the Attainable Region for Batch Reactor Networks

Benjamin J. Davis, Larry A. Taylor, and Vasilios I. Manousiouthakis*

Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California 90095

In this work, we describe a method for automatically identifying the set of all points in concentration space that represent outlet compositions of some network of discretely fed batch reactors for a given reaction set with known kinetics. This so-called batch attainable region (BAR) is dependent on the batch network’s feed and total operating time, and it is shown to be quantifiable using the Infinite DimEnsionAl State-space (IDEAS) framework. We first establish that a simple batch reactor model possesses the properties that allow application of the IDEAS framework. We then formulate the resulting IDEAS Infinite Linear Program (ILP) whose solution is guaranteed to identify the globally optimal network of batch reactors. We subsequently use a simple transformation of this IDEAS ILP that leads us to propose two algorithms that are related to the construction of the true BAR. The first is a “Shrink-Wrap”-like algorithm that is similar to that previously reported [Manousiouthakis et al. The Shrink-Wrap Algorithm for the Construction of the Attainable Region: Application of the IDEAS Framework. Comput. Chem. Eng. 2004, 28, 1563] and creates increasingly accurate approximations of a set guaranteed to contain the true BAR for all network operating times. The second is a breadth-first algorithm that creates increasingly accurate inner approximations to the BAR for a given network operating time. These two algorithms are applied to an example from the literature and are shown analytically to converge in the limit to the true BAR.

Introduction

Evaluation of limits on the performance of reactors and reactor networks is crucial to the economic success of any chemical process network. Consequently, the analysis and design of reactors and reactor networks have been the primary foci of process systems engineering research. Previous works on non-steady-state reactor networks have addressed techniques for analyzing, modeling, or optimizing single-batch units, but they make little mention of how units can be used in conjunction or what theoretical limits exist on the performance of these types of non-steady-state systems. Because non-steady-state networks are fundamentally different from their steady-state counterparts, identification of these performance limits requires careful consideration of the effect of time (both reaction and holding time) and causal relationships between reactors.

The goal of this work is to apply the Infinite DimEnsionAl State-space (IDEAS) framework to construction of the batch attainable region (BAR) for non-steady-state networks of batch reactors; this is the first application of IDEAS to a network of dynamic process units. The remainder of the work is structured as follows: first, we give background information on reactor network synthesis (RNS), the IDEAS conceptual framework, and attainable region (AR) construction. Next, the applicability of IDEAS to batch RNS is established and the relevant IDEAS infinite linear program (ILP) is formulated. We then outline a variable transformation which leads to two algorithms that are related to the construction of the true BAR: the first is similar to the “Shrink-Wrap” algorithm that was developed by Manousiouthakis et al. for the construction of the steady-state AR and creates increasingly accurate approximations of a set guaranteed to contain the true BAR for all network operating times. The second is a breadth-first algorithm that creates increasingly accurate inner approximations to the BAR for a given network operating time. These two algorithms are applied to an example from the literature and are shown analytically to converge in the limit to the true BAR.

Background

Automatic (computer-based) reactor network synthesis (RNS) evolved as a field of its own starting in the 1980s. Chitra and Govind performed work in 1981 on the identification of optimal reactor types and configurations using a super-structure-based approach. They later expanded on their earlier work, applying a superstructure-based method for optimal RNS for both isothermal and non-isothermal reaction systems. Ong considered the optimization of continuously stirred tank reactors (CSTRs) in series using Bellman’s dynamic programming. Pibouleau et al. proposed a mixed-integer nonlinear programming (MINLP) formulation for the automatic synthesis of networks featuring CSTRs and single-stage separation units. In 1994, Omtveit et al. presented a RNS method that also included a separation network as a separate sub-problem; paper also gives an extensive literature review of work on RNS to that point. Smith and Pantelides gave another reactor—separator network superstructure-based formulation in their 1995 work. Bikic and Glavic produced a series of papers on a superstructure-based nonlinear programming (NLP) method for RNS for networks with multiple multicomponent feeds, non-isothermal complex reaction schemes, and reactor/separators networks. Their 1996 work stated that “the proposed design procedure can also be used to support the design of batch processes”, but that work did not specifically address the claim. Esparta et al. proposed a superstructure-based method for RNS using isothermal two-phase CSTRs in 1998. Hua et al. proposed a NLP model for RNS that included “differential recycling DSR” reactor units. Mehta and Kokossis proposed a stochastic optimization approach to non-isothermal and multiphase RNS. Pahor et al. proposed a superstructure/MINLP approach for optimization and then, in later work, applied the method to the non-isothermal production of allyl chloride. Moreover, in 1999, Grossman et al. gave a review of advances in mathemati-
cal programming for process systems synthesis; this review has sections devoted to both RNS and AR construction theory, respectively.

The RNS problem was also approached using the methods of optimal control; the work by Aris was significant in this field, addressing the problem of optimal control of a batch reactor. In 1970, Paynter and Haskins formulated an optimal control problem for determining the optimal reactor type for a single reactor using an axial dispersion model. Waghmare and Lim applied optimal control theory to single isothermal reactor systems and later applied the same techniques to complex reaction schemes. Achenie and Biegler proposed an NLP method for RNS that used superstructures in a “target-based” approach. Goddor et al. outlined optimal control policies for reactor structures on the AR boundary, using temperature as the control variable. Hillestad formulated the RNS problem as an optimal control problem and then examined its solution for the isothermal and non-isothermal cases.

The IDEAS conceptual framework was proposed by Manousiouthakis et al. in an effort to overcome two limitations of superstructure-based optimal process network synthesis methods: (i) the considered superstructure may impose unforeseen limitations on the eventually obtained optimal network, and (ii) the nonconvex nature of the resulting superstructure-based optimization formulations (NLP, MINLP, etc.) only guarantees local optimality of the obtained optimal network. IDEAS overcomes these limitations by considering all possible process network configurations and establishing that most commonly applied process models can be used to yield optimization formulations with an infinite number of variables and an infinite number of linear constraints. The ability of IDEAS to address several long-standing process network synthesis problems has been demonstrated on the minimum utility cost (MUC) problem for mass exchange networks (MENs), the minimum plate area and MUC problems for heat-integrated complex distillation networks, the minimum total annualized cost (MTAC) problem for separation networks and power cycle networks, the MUC problem for heat and power integrated complex distillation networks, and the minimum total liquid hold-up problem for complex reactive distillation networks.

The attainable region (AR) for a given set of reactions and reactor technologies is defined as the set of all points in concentration space that are attainable through reaction and mixing from a given feed point; this definition has been widely credited to Horn in 1964. Quantification of the AR for reactor mixing from a given feed point; this definition has been widely.

In 1997, Feinberg and Hildebrand reported on the determination of the optimal reactor network configuration using the geometric properties of the AR extreme points in the first part of a three-part series. This work was later expanded by Feinberg in 1999 to a more extensive set of mathematical properties of the AR boundary and in 2000 to properties of critical DSRs and CSTRs whose outlets are on the AR boundary. In 1999, McGregor et al. examined the relationship between the geometric AR identification method and Pontryagin’s maximum principle. Rooney et al. extended the geometric AR identification method to higher dimensions by extending the 2D subspace. In 1997, Nisoli et al. outlined a method for identifying the AR for a two-phase reaction—separation system; they applied the method to the production and separation of dimethyl ether (DME) from methanol and methyl tert-butyl ether (MTBE) from isobutene and methanol. That same year, Smith and Malone outlined an application of AR identification in the free-radical polymerization of poly(methyl methacrylate) (PMMA). Later, in 2002, Gadewar et al. analyzed networks of two-phase CSTRs that are surrogates for reactive distillation units to find an AR for such networks.

Kauchali et al. used the earlier methods of Nisoli et al. to identify candidate ARs for the water-gas shift (WGS) reaction, which is a problem that also was previously studied by Omtev et al.

More recently, the IDEAS conceptual framework has also been applied in the construction of the AR for reactor networks. Burri et al. first presented several IDEAS-based infinite linear programming (ILP) formulations of the AR construction problem in 2002. That same year, Kauchali et al. independently developed an IDEAS-like linear programming model for extending candidate ARs. Manousiouthakis et al. presented properties of one of the aforementioned IDEAS ILPs, which allowed construction of the true AR without explicit solution of the ILP using a so-called “Shrink-Wrap” algorithm. Concurrent and independent work by Abraham and Feinberg proposed a method of bounding hyperplanes to identify subsets of a superset containing the AR that are guaranteed not to contain the AR. More recently, Zhou and Manousiouthakis demonstrated that variants of the Shrink-Wrap algorithm are applicable to the AR construction for nonideal axial dispersion reactor models and variable-density fluid reactor models, respectively.

There is extensive work in the literature that addresses single-batch reactor optimization and batch process scheduling. Ripp et al. gave a review of studies of individual batch units and their optimization in 1983. That same year, he also wrote an overview of general structures for batch process systems. He followed up these reviews 10 years later with the current progress in the field of engineering and design of batch processes. Reklaitis gave a review of progress and issues in computer-aided batch process design in 1990. Leven et al. wrote on the optimal design of batch, discrete semi-batch, and continuous semi-batch reactor units in his 1992 work. Terwiesch et al. surveyed industry needs for batch processes and suggested both optimal control methods for improvements and also further research problems to be addressed. Yi and Reklaitis have performed work on the optimal synthesis of batch storage networks for chemical processes. Maravelias examined the problem of optimal scheduling of single-stage and multistage batch processes using a mixed-integer sequencing algorithm. Later work by Sung and
Maravelias defines a “process attainable region” for production planning and scheduling problems with limited equipment capacity.

Application of IDEAS to Batch Reactor Network Synthesis

The fundamental difference between batch and steady-state reactor network synthesis problems is the time dependence of the underlying batch reactor process. In a network context, this time dependence immediately raises the issue of “causality”; in a batch reactor network, a reactor cannot “feed” another reactor unless its contents are unloaded before the other reactor’s loading time. To examine causality in a straightforward manner, we consider that batch reactor loading, unloading, and mixing operations can only occur at prespecified discrete times, \( t_0, t_1, t_2, ..., t_n \) and are instantaneous. For the remainder of this work, we will refer to the reactor inlet and outlet volumes (which are equal, because of our assumption of constant density) as flow rates; this is a slight misnomer, because each reactor’s inlet and outlet are discrete (i.e., there is no temporal nature to the flow into or out of a reactor). But the convention of calling them flow rates makes the analogy between batch reactor networks and other steady-state reactor networks clearer. The inlet and outlet volumes of a batch reactor unit are extensive properties that do not affect the units’ intensive properties, such as species concentrations, temperature, pressure, etc.

The standard model for a batch reactor is outlined in almost any reaction engineering textbook (see the work of Levenspiel, Froment and Bischoff, Schmidt, Fogler, Nauman, Rawlings, etc.). More recently, there have been more-complicated batch reactor models proposed in the literature that account for imperfect mixing and continuously fed batch operation; however, we will not specifically address these complications in this work. To apply IDEAS to the batch reactor network synthesis problem, we must first prove that the batch reactor model is flow-invariant with respect to its intensive properties. Second, we must prove that the operations of mixing and splitting in the distribution network are linear. A more formal mathematical method for showing the applicability of IDEAS to a unit model has been previously outlined in Zhou et al. We define our IDEAS-inlet-outlet information map \( M \) for the batch reactor model as follows:

\[
M: u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \rightarrow y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} M_1(u_2)u_1 \\ M_2(u_2) \end{bmatrix} \tag{7}
\]

We will define these vectors as follows:

\[
u_1 = [\bar{F}] = [F] = y_1 \tag{8}
\]

\[
u_2 = [\bar{C}(\bar{C}) \bar{C}(2) \ldots \bar{C}(N) \ t_{in} \ t_{out} \ \lambda] \tag{9}
\]

\[
y_2 = [C(1) C(2) \ldots C(N)] \tag{10}
\]

Because \( y_1 = u_1 \), the map \( M_1 \) is just the identity map. We uniquely define the map \( M_2 \) by assuming in this work that the set of differential equations in eq 2 with initial conditions in eq 3 admits a solution and that the solution is unique. Sufficient conditions on the properties of the rate vector for this to be true can be found in theorem 2.4 of Khalil. Each of the outlet concentrations can be found using the information in \( u_2 \); therefore, we can define the map \( M_2 \) as follows:

\[
M_2: u_2 \xrightarrow{eqs \text{ 2-6}} y_2
\]

Therefore, this batch reactor model satisfies the first necessary property for the applicability of the IDEAS framework. The second property—that the operations of mixing, splitting, and recycle are linear—is obvious, given that the intensive properties of the units are considered to be known.

Graphical Representation of Batch Reactor Networks

As an example, consider a simple batch reactor network that consists of three reactors and two holding tanks. Reactor 1 is fed at some initial time (we will call it \( t_0 \)) and runs until its outlet time \( t_1 \). At that time, the reactor 1 outlet is split into three parts, which are fed into reactor 2, reactor 3, and holding tank 1, respectively. Reactor 2 operates from \( t_1 \) (which is the output time of reactor 1) until time \( t_2 \) and then its outlet enters holding tank 2. Reactor 3 operates from \( t_1 \) until \( t_3 \), at which time its outlet is mixed with the contents of holding tanks 1 and 2 to form the network outlet. This simple batch reactor network can be visualized in a time-axis form, as shown in Figure 1.

We can also represent this same process flowsheet in an OP/DN form, by rearranging the process network diagram such that all the mixing and splitting operations are contained in the distribution network (DN) and all unit operations (the three batch reactors and two holding tanks) are contained in the operator block (OP). This method of representing a flowsheet can be used to best visualize the IDEAS formulation of the batch reactor network problem. In the IDEAS formulation, all possible reactors are considered in the formulation of the optimization problem; therefore, instead of a diagram with three reactors, there is an infinite number of batch reactors and holding tanks.
The inlet time is arbitrarily designated as \( t_0 \). Thus, \( n \) designates the number of times that the reactor/holding tank contents are allowed to mix (\( n = 3 \) in the example above). Also note that holding tanks are implicitly included in this formulation as units in the OP block (no holding occurs in the DN).

### IDEAS ILP Formulation for Batch Reactor Network Synthesis

The IDEAS optimization problem outlined below considers every possible reactor and holding tank unit and all causally feasible interconnections in the DN, thereby guaranteeing that all possible batch reactor networks are included in the mathematical problem formulation. Without any loss of generality, each unit’s input information vector \( \mathbf{x}_2 \) is considered to be known, and thus, using the information map \( \mathbf{M} \), its output information vector \( \mathbf{y}_2 \) is also known. Knowledge of each unit’s input and output information vectors \( \mathbf{y}_2 \) and \( \mathbf{x}_2 \) will be shown below to lead to an IDEAS formulation of the optimal batch reactor network synthesis (RNS) problem that is an infinite linear program (ILP). Before we can outline this IDEAS ILP, we first define (infinite) sets of reactors that are organized by their inlet and outlet times:

\[ S_i^j \equiv \text{the set of all reactors and holding tanks that operate from time } t_i \text{ to time } t_j \] (11)

Using these sets as defined, the formulation for the batch reactor network synthesis problem can be set up as follows:

\[ \text{minimize } v = f(\mathbf{F}^{0k}_{m}, \mathbf{F}^{0k}_{j}, \mathbf{F}^{ij}_{m}, \mathbf{F}^{ij}_{j}, \mathbf{F}^{0k}_{m}, \mathbf{F}^{0k}_{j}) \]

subject to:

1. Distribution Network Inlet Mass Balance (Splitting)

\[ \mathbf{F}_* = \sum_{k=1}^{n} \sum_{m \in S_i^k} \mathbf{F}^{0k}_{m} = \sum_{k=1}^{n} \sum_{m \in S_i^k} \mathbf{F}^{0k}_{m} = \sum_{k=1}^{n} \sum_{m \in S_i^k} \mathbf{F}^{0k}_{m} \]

2. OP Outlet Mass Balances (Splitting)

\[ \mathbf{F}^{ij}_{m} = \sum_{k=2}^{n} \sum_{l \in S_i^k} \mathbf{F}^{ij}_{lm} \]

\[ \forall m \in S_i^j; \forall i = 0, ..., n - 2; \forall j = 1, ..., n - 1 (i < j) \]

3. Distribution Network Outlet Mass Balance (Mixing)

\[ \mathbf{F}_* = \sum_{i=0}^{n-1} \sum_{l \in S_i^j} \mathbf{F}^{in}_{l} = \sum_{i=0}^{n-1} \sum_{l \in S_i^j} \mathbf{F}^{in}_{l} \]

4. Distribution Network Outlet Component Balances (Mixing)

\[ \mathbf{C} \cdot \mathbf{F}_* = \sum_{i=0}^{n-1} \sum_{l \in S_i^j} \mathbf{C}^{in}_{l} \mathbf{F}^{in}_{l} = \sum_{i=0}^{n-1} \sum_{l \in S_i^j} \mathbf{C}^{in}_{l} \mathbf{F}^{in}_{l} \]

5. OP Inlet Mass Balances (Mixing)

\[ \mathbf{F}^{0k}_{m} = \sum_{i=0}^{n-2} \sum_{l \in S_i^j} \mathbf{F}^{ij}_{lm} \]

\[ \forall m \in S_i^j; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k) \]
(6) OP Component Balances (Mixing)
\[
\overline{C}_{mi} F^j_{ml} = \sum_{i=0}^{n-2} \sum_{i=j}^{k} C^j_{mi} F^j_{ml}
\]
\(\forall m \in S^k, \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k)\)

(7) Positivity Constraints
\[
\overline{F}^j_{ml} \geq 0, \quad F^i_{ml} \geq 0, \quad F^j_{lm} \geq 0 \quad \forall m \in S^j; \forall l \in S^k;
\]
\(\forall i = 0, ..., n - 2; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (i < j < k)\)

(8) Constant Density Assumption
\[
F^j_{ml} = \overline{F}^j_{ml} - F^i_{ml} \quad \forall m \in S^j; \forall i = 0, ..., n - 1; \forall j = 1, ..., n
\]

(9) Batch Reactor/Holding Tank Model Map
\[
\begin{align*}
[\overline{C}^{j}_{m}(1) \quad C^{j}_{m}(2) \quad ... \quad \overline{C}^{j}_{m}(N)] & \quad t_i \quad t_j \quad \lambda_{ij}^1 \quad M_z \\
[C^{j}_{m}(1) \quad C^{j}_{m}(2) \quad ... \quad C^{j}_{m}(N)] & \quad \forall m = 1, ..., \infty; \forall i = 0, ..., n
\end{align*}
\]
if \(\lambda_{nj}^1 = 1\): Batch Reactor, \(R_j(\overline{C}^{j}_{m}(i)) \neq 0\) \(\forall i = 0, ..., N\)

if \(\lambda_{mj}^1 = 0\): Holding Tank, \(R_j(C^{j}_{m}(i)) = 0\) \(\forall i = 0, ..., N\)

The aforementioned optimization problem is an ILP (because all unit inlet and outlet concentrations are known) as long as the objective function \(f\) is a linear function of the network’s “flow” variables. The most commonly considered linear objective functions are maximization or minimization of one particular component’s DN outlet concentration, maximization of selectivity or yield of a desired product, minimization of total network volume, etc.

In practice, this ILP is not solved directly; this would be impossible except for certain specific cases where a solution could be found analytically in the limit. Instead, finite linear programs (or LPs) are solved that give an approximation to the ILP’s solution. As larger and larger LPs are solved, their solutions more accurately approximate the solution of the ILP. If we represent the number of units considered in our LP approximation as \(U\) and the associated objective function value for the LP as \(v_{U}\), then the infinite sequence
\[
\{v_U\}_{U=1}^{\infty}
\]
has been proven to be a non-increasing sequence of upper bounds on the actual ILP objective function value \(v_{\infty}\) (or just \(v\) and the sequence is guaranteed to converge to \(v\) (proof of this is given in work by Justanieah, in collaboration with Manousiouthakis).

**Definition of Mixing Ratios**

Now that we have defined our problem above, we will introduce the concept of a mixing ratio, \(a_{lj}^i\). A mixing ratio is a ratio between 0 and 1 (inclusive) that defines the fraction of a particular unit’s inlet stream (or the network outlet) coming from another unit (or from the network inlet). Mixing ratios satisfy the following equations:
\[
a_{lj}^i \geq 0, \quad a_{lj}^i \leq 1
\]
\(\forall m \in S^k, \forall l \in S^j; \forall j = 0, ..., n - 2; \forall k = 2, ..., n (i < j < k)\)

Similar equations can be written for the OP inlet and network outlet mixing ratios:
\[
a^{i}_{ml} = \overline{a}^{i}_{ml} \quad \forall m \in S^j; \forall j = 0, ..., n - 1; \forall k = 2, ..., n (j < k)
\]

To identify the BAR, we then introduce the following definitions:

**Definition 1:** Active unit – a unit (reactor or holding tank) is active if its volume (outlet flow rate) is nonzero (strictly positive) in the feasible solution to the above optimization problem.

**Definition 2:** Inactive unit – a unit is inactive if its volume is zero in the feasible solution to the above optimization problem.

Without any loss of generality, mixing ratios from inactive to active units can be set to zero, because flows to or from an inactive unit are zero. In addition, the sum of all mixing ratios that correspond to any unit inlet can be set to one. Indeed, for mixing ratios that correspond to active unit inlets, this must be the case, whereas for mixing ratios that correspond to inactive units, this assumption is inconsequential. Furthermore, constraint set 6 of the aforementioned IDEAS ILP is automatically satisfied for an inactive unit, while it suggests that the inlet concentration vector of an active unit fed at time \(t_i\) belongs to the convex hull of outlet concentration vectors of active units, which outlet at time \(t_j\).

We also consider that these mixing ratios satisfy the following five conditions; these equations (eqs 18–22) are automatically satisfied for active units and are irrelevant to the IDEAS ILP formulation for units with zero flow, based on constraint sets 3, 5, and 6 in the above formulation:
\[
\sum_{i=0}^{n-2} \sum_{l<j<k} a_{lj}^i = 1
\]
\(\forall m \in S^k; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k)\) (18)

\[
\sum_{i=0}^{n-1} \sum_{l<j<k} a_{lj}^i = 1
\]
(19)

\[
\sum_{i=0}^{n-2} \sum_{l<j<k} \overline{C}^{j}_{m} a_{lj}^i = \overline{C}^{j}_{m}
\]
\(\forall m \in S^k; \forall j = 0, ..., n - 2; \forall k = 2, ..., n (j < k)\) (20)

\[
a_{lj}^i = 1 \quad \forall m \in S^k; \forall k = 1, ..., n
\]
(21)

\[
\overline{C}^{j}_{m} = C \quad \forall m \in S^k; \forall k = 1, ..., n
\]
(22)

We can substitute in these mixing ratios (eqs 13–17 above) into the feasible set of the IDEAS ILP to transform it to an
equivalent set. We also make use of the constant density assumption to eliminate some of the redundant flow variables and positivity constraints, reforming the set as follows:

1. \[ F_m = \sum_{k=1}^{n} \sum_{i \in S^k} F_m^{ik} \phi_m^{ik} = \sum_{k=1}^{n} \sum_{i \in S^k} F_m^{ik} \]

2. \[ F_m^i = \sum_{k=2}^{n} \sum_{l \in S^l} F_m^{ij} d_{ml}^{ij} \]

\[ \forall m \in S^k; \forall i = 0, ..., n - 2; \forall j = 1, ..., n - 1 \] (i < j)

3. \[ 1 = \sum_{i=0}^{n-1} \sum_{l \in S^l} \phi_{il}^{in} \]

4. \[ \bar{C}_m = \sum_{i=0}^{n-1} \sum_{l \in S^l} C_{il}^{in} \phi_{in} \]

5. \[ 1 = \sum_{i<j}^{n-2} \sum_{l \in S^l} d_{ml}^{ij} \]

\[ \forall m \in S^k; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k) \]

6. \[ \bar{C}_m^{ik} = \sum_{i<j}^{n-2} \sum_{l \in S^l} C_{il}^{in} d_{ml}^{ij} \]

\[ \forall m \in S^k; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k) \]

7. \[ F_m^i \geq 0, \quad 0 \leq d_{ml}^{ij} \leq 1 \]

\[ \forall m \in S^k; \forall l \in S^l, \forall i = 0, ..., n - 2; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (i < j < k) \]

\[ 0 \leq \phi_{in} \leq 1 \]

\[ \forall i = 0, ..., n - 1; \forall k = 1, ..., n; \forall m \in S^0; \forall l \in S^m \]

We have omitted the batch reactor model equations, because it is assumed that the infinite sequence of reactor inlet concentrations and their respective lambda parameters are known for a given reactor’s inlet and outlet time, and thus all outlet concentrations are also known for each unit using the input–output map \( M \). Also, this formulation assumes the positivity of the network inlet flow, because the problem would be trivial otherwise. All the constraints in the aforementioned set are linear, except for constraint set 2 (which corresponds to the OP outlet splitting mass balances).

**Finite Approximation to Feasible Set**

To approximately identify the solution to the aforementioned infinite problem, we discretize the concentration space with a finite number of grid points. We then consider all units with an inlet composition vector at one of the grid points, some at starting time \( t_s \), and some at ending time \( t_e \). Let \( U \) be the finite number of units that operate from each starting time \( t_s \) to each ending time \( t_e \); we will define new finite sets \( G^j \) with cardinality \( U \), which approximate the infinite sets \( S^j \). The vector of all flows through each of these units in set \( G^j \) can be defined as follows:

\[ F^j = [F^j_1 F^j_2 F^j_3 \cdots F^j_U]^T \]

\[ \forall i = 0, ..., n - 1; \forall j = 1, ..., n (i < j) \] (23)

**Figure 3.** Matrix \( A \) from eq 24.

The previously mentioned constraint sets 1 and 2 then lead to the matrix equation: where \( A \) is a matrix of mixing ratios (shown in Figure 3) and \( \mathbf{F} \) is given as:

\[ \mathbf{A} \mathbf{F} = \mathbf{F} \]

(24)

\[ \hat{A} = \begin{bmatrix}
0 & A_{11} & \cdots & A_{1n} & 0 & \cdots & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & A_{21} & \cdots & A_{2n} & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & A_{n-1,n} \\
\end{bmatrix} \]

The sub-matrices \( A_{ij} \) in \( A \) contain some combination of mixing ratios and zeros. The matrices \( A_{y1}, A_{y2}, \ldots, A_{yn} \) contain only ones (see eq 21).

Equation 24 (\( \mathbf{A} \mathbf{F} = \mathbf{F} \)) must be satisfied for any feasible solution to the new finite problem, under the condition that \( U \) is large enough that at least one feasible solution exists. Assuming that the units in each set \( G^j \) can be rearranged in the proper manner, the column sums of this matrix must satisfy eqs 18, 19, and 21 for the mixing ratios to be feasible for the problem. Because these mixing ratios are between 0 and 1 inclusive, the matrix \( \mathbf{A} \) is a non-negative square matrix with column sum of 1. Bapat and Raghavan\(^7\) have shown that this result proves that the spectral radius of \( \mathbf{A} \) is also 1, using the Perron–Frobenius Theorem (Lemma 3.1.1). This implies that there exists an eigenvector of the matrix \( \mathbf{A} \) that is \( \geq 0 \) and with a corresponding eigenvalue of 1 (see theorem 1.7.3 in the work of Bapat and Raghavan\(^7\)). Therefore, if a matrix \( \mathbf{A} \) of mixing ratios can be found that is feasible for the original optimization problem, one can always find a vector \( \mathbf{F} \) of feasible flows such that \( \mathbf{A} \mathbf{F} = \mathbf{F} \) and \( \mathbf{F} \geq 0 \). Using this result, the necessary conditions for a point to be in the feasible set of the transformed problem can be modified, with constraint sets 1, 2, and part of set 7 eliminated, leaving:

3. \[ 1 = \sum_{i=0}^{n-1} \sum_{l \in S^l} \phi_{in} \]

4. \[ \bar{C}_m = \sum_{i=0}^{n-1} \sum_{l \in S^l} C_{il}^{in} \phi_{in} \]

5. \[ 1 = \sum_{i=j}^{n-2} \sum_{l \in S^l} d_{ml}^{ij} \]

\[ \forall m \in S^k; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k) \]
6. \[ C_m^{jk} = \sum_{i=0}^{n-2} \sum_{l \in S_i^j} C_i^j k l \]
\[ \forall m \in S^k, \forall j = 1, ..., n - 1; \forall k = 2, ..., n (j < k) \]

7. \[ 0 \leq d_{lm}^{jk} \leq 1 \]
\[ \forall m \in S^j; \forall l \in S^k; \forall i = 0, ..., n - 2; \forall j = 1, ..., n - 1; \forall k = 2, ..., n (i < j < k) \]
\[ 0 \leq d_{ij}^{lm} \leq 1 \]
\[ \forall i = 0, ..., n - 1; \forall k = 1, ..., n; \forall m \in S^k; \forall l \in S^n \]

Constraint 4 can be thought of as a specification on the outlet concentration for the feasible solution. The set of mixing ratios that satisfy the aforementioned constraints defines the AR for the batch reactor network for a given \( n \); this feasible set is similar to that obtained in Manousiouthakis et al. for steady-state RNS. Because of the fact that there is only one network inlet, there can only be one possible inlet concentration set for each reactor/holding tank that is fed at time \( t_0 \) (the feed concentration). Similarly, all reactors fed at time \( t_1 \) can only be fed by the effluent of the reactor with inlet time \( t_0 \) and outlet time \( t_1 \) or the holding tank with inlet time \( t_0 \) and outlet time \( t_1 \). Reactors that feed at time \( t_2 \) can be fed by some linear combination of the effluents of reactors/holding tanks that outlet at \( t_2 \) and so on. Generally, to ensure causality, a reactor or holding tank that feeds at \( t_i \) can be fed only by the linear combination (mixing) of effluents of reactors that outlet at \( t_i \). The fact that the lower triangle (excluding the first column) of matrix \( A \) (Figure 3) is zero is a manifestation of the network’s causality, namely that unit feeds at time \( t_i \) can only be comprised of unit effluents at time \( t_i \).

**Shrink-Wrap BAR Construction Algorithm**

An algorithm that is computationally similar to the “Shrink-Wrap” algorithm (outlined in Manousiouthakis et al.) can be applied to the construction of a superset of the true BAR. Because of the causal nature of the batch RNS problem, this method can only be guaranteed to find a superset to the true BAR. One of the steady-state RNS problem characteristics that leads to the Shrink-Wrap steady-state AR construction method is that any unit’s feed can be constructed, through mixing, as a linear combination of the feed and reactor unit exits that belong to the candidate steady-state AR. This was due to the fact that all points in the candidate AR could be constructed independently of each other (through mixing) without consideration of “when” each point was generated. Because unit outlets in the batch reactor network case cannot be used arbitrarily to reconstruct other unit inlets (because of causality), the availability of any arbitrary unit exit in the candidate BAR to mix and convexify the candidate BAR cannot be guaranteed unless one knows when that unit operates. However, a Shrink-Wrap-like algorithm can be used to identify increasingly accurate approximations of a convex set that is guaranteed to contain the BAR for all network operating times \( n \), although it is not guaranteed to necessarily converge to the true BAR. This algorithm removes extreme points (vertices) from an initial superset by following the batch reactor trajectory backward (for a given time, \( t_1 - t_0 \)) and evaluating from where it started. If the starting point of the vertex’s trajectory is not a point in the interior of the superset, then no batch reactor can exist that creates that point and the vertex is removed. The algorithm then updates the set as points are removed. The method essentially eliminates points in concentration space (identifying them as “unattainable”) by explicitly calculating from where each point must have come. An explicit description of this algorithm is given as follows:

1. Identify a suitable superset in concentration space for the system in question, based on knowledge of its physical constraints.
2. Discretize the superset in all directions; the level of accuracy of the method is improved as this discretization becomes finer.
3. Start from an extreme point of the current set and travel backward along the path of the batch reactor in concentration space from \( t_1 \) to \( t_0 \). If this new point is in the current set, keep it. If it is not, remove it from the set.
4. Repeat step 3 until no more extreme points can be removed; this region is guaranteed to contain the true BAR for all network operating times \( n \) and for a given reaction/holding time, \( t_1 - t_0 \).

**Breadth-First BAR Construction Algorithm**

The aforementioned Shrink-Wrap-like algorithm is only guaranteed to converge to a superset of the BAR. An alternative algorithm that is guaranteed to converge to BAR_0 (for any given \( n \)) is given next. BAR_0 is “grown” from time \( t_0 \) out to time \( t_n \) to respect the time hierarchy of the unit operations in a forward dynamic programming-like manner. The algorithm is an instance of a breadth-first search (see the work of Cormen et al.) with the network feed as the first point in concentration space in the candidate BAR. At each stage of generation, the points or nodes form a directed acyclic graph (DAG). Because of this, it is easy to construct an actual network of operations from the nodes of the graph. Each node of the DAG, other than the first, can either have a single parent or multiple parents, because of mixing. If the node is generated by a batch reactor simulation, it has only one parent and that batch reactor operation will be added to the resulting network of reactors to create that node. If the node is generated by a linear combination of other nodes (mixing), it will have two or more parents. At least one of the parent nodes is a newly generated vertex from the previous time step, because we have assumed that mixing is instantaneous. An outline of the algorithm is given as follows:

1. Start from the feed point—this is the BAR at time \( t_0 \) or BAR_0.
2. Travel along the path of the batch reactor in concentration space from \( t_0 \) to \( t_1 \). This will be the exit concentration vector of the first batch reactor.
3. Find the convex hull of all the points from the previous time step and the points from the new time step to find the BAR at that particular time. For the first time step, this will be a line. This is BAR_1.
4. Discretize this line and travel from time \( t_1 \) to time \( t_2 \) along the reactor trajectories starting at all discretized points on the line BAR_1 that are not part of BAR_0.
5. Form the convex hull of this new set and BAR_1; this is an approximation of BAR_2.
6. For every BAR_m, (a) generate the reactor trajectories from time \( t_i \) to time \( t_{i+1} \) for all points of BAR_0 on an appropriately defined grid that are not contained in BAR_{m-1}, and (b) form the convex hull of this new set of points with the approximation of BAR (this is an approximation of BAR_2).
7. Repeat step 6 until time \( t_n \); this final convex hull is an approximation of BAR_n, the true BAR at \( t_n \).
Table 2. List of Points Used in Forming BAR0, BAR1, BAR2, BAR3, and BAR

<table>
<thead>
<tr>
<th>point</th>
<th>first generated in</th>
<th>C(A)</th>
<th>C(C)</th>
<th>formed by</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>BAR2</td>
<td>0.2486</td>
<td>0.3204</td>
<td>mixing</td>
</tr>
<tr>
<td>B</td>
<td>BAR2</td>
<td>0.1667</td>
<td>0.3493</td>
<td>map of point C</td>
</tr>
<tr>
<td>C</td>
<td>BAR1</td>
<td>0.375</td>
<td>0.2466</td>
<td>map of point F</td>
</tr>
<tr>
<td>D</td>
<td>BAR3</td>
<td>0.0625</td>
<td>0.3931</td>
<td>map of point B</td>
</tr>
<tr>
<td>E</td>
<td>BAR3</td>
<td>0.2158</td>
<td>0.3495</td>
<td>mixing</td>
</tr>
<tr>
<td>F</td>
<td>BAR0</td>
<td>1.0</td>
<td>0.0</td>
<td>network feed point</td>
</tr>
<tr>
<td>G</td>
<td>BAR3</td>
<td>0.1064</td>
<td>0.3882</td>
<td>map of point A</td>
</tr>
<tr>
<td>H</td>
<td>BAR</td>
<td>0.0</td>
<td>0.1851</td>
<td>see Appendix C</td>
</tr>
<tr>
<td>W</td>
<td>BAR</td>
<td>0.05752</td>
<td>0.4568</td>
<td>map of point Z</td>
</tr>
<tr>
<td>Y</td>
<td>BAR</td>
<td>0.0</td>
<td>0.4666</td>
<td>see Appendix D</td>
</tr>
<tr>
<td>Z</td>
<td>BAR</td>
<td>0.1579</td>
<td>0.4156</td>
<td>see Appendix D</td>
</tr>
</tbody>
</table>

Case Study for Trambouze Kinetics

We will apply the two aforementioned algorithms to the construction of the BAR for a case study that exhibits Trambouze's reaction kinetics (example 4 from his 1959 work), which is a reaction scheme that is often studied in the literature.\(^1\)\(^3\)\(^8\)\(^55\)

\[ k_1 \text{ A} \rightarrow \text{ B (zeroth order, } k_1 = 0.025 \text{ mol L}^{-1} \text{ min}^{-1}) \]  
(26)

\[ k_2 \text{ A} \rightarrow \text{ C (first order, } k_2 = 0.2 \text{ min}^{-1}) \]  
(27)

\[ k_3 \text{ A} \rightarrow \text{ D (second order, } k_3 = 0.4 \text{ L mol}^{-1} \text{ min}^{-1}) \]  
(28)

These are the same reaction kinetics and parameters as those given in the work of Manousiouthakis et al.\(^1\) The rates of generation for each species for this reaction pathway are

\[ R_A = -k_1 - k_2 C_A - k_3 C_A^2 \]  
(29)

\[ R_B = k_1 \]  
(30)

\[ R_C = k_2 C_A \]  
(31)

\[ R_D = k_3 C_A^2 \]  
(32)

The batch reactor equations can be solved analytically, as described in Appendix A. This makes this example ideal for testing our candidate BAR\(_n\) identification algorithms, because they can be compared analytically to the true BAR\(_n\) at each time. For the purpose of creating BAR\(_n\)s for this example, we consider that the batch reactor time is 2 min, i.e., that \( t = t_n - t_{n-1} = t_{n-1} - t_{n-2} = \ldots = t_1 - t_0 = 2 \text{ min} \).

For this batch reactor time, BAR\(_0\), BAR\(_1\), BAR\(_2\), and BAR\(_3\) are created analytically for this example as described by Appendix B and BAR (BAR\(_n\) as \( n \rightarrow \infty \)) is identified as

\[ f_{i}(C(A)) = C(C) = 0.8(0.2 - \ln 2)(C(A) - 1) \]  
(33)

\[ g_{i}(C(A)) = C(C) = (0.2 - \ln 2)8C(A) - 34 - 4C(A) + \ln \left( \frac{5}{4 - 4C(A)} \right) - 0.1 \]  
(34)

\[ h_{i}(C(A)) = C(C) = 0.426\left( \frac{15 - 4C(A)}{16 - 16C(A)} \right) + \ln \left( \frac{5}{4 - 4C(A)} \right) - 0.1 \]  
(35)

\[ j_{i}(C(A)) = C(C) = 0.2(0.2 - \ln 2)12C(A) - 23 - 8C(A) + \ln \left( \frac{5}{3 - 8C(A)} \right) - 0.2 \]  
(36)

\[ j_{i}(C(A)) = C(C) = \left( \frac{40C(A) - 15}{16 - 16C(A)} \right)C(A) + 0.5 \ln \left( \frac{5}{4 - 4C(A)} \right) - 0.1 \]  
(37)

\[ j_{i}(C(A)) = C(C) = \left( \frac{30C(A) - 5}{6 - 16C(A)} \right)C(A) + 0.5 \ln \left( \frac{5}{3 - 8C(A)} \right) - 0.2 \]  
(38)

Table 3. List of Points and Relevant Equations for Curves That Define BAR\(_0\), BAR\(_1\), BAR\(_2\), BAR\(_3\), and BAR

<table>
<thead>
<tr>
<th>region</th>
<th>defined by</th>
<th>relevant equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAR(_0)</td>
<td>point F</td>
<td>( F = (1,0,0,0) )</td>
</tr>
<tr>
<td>BAR(_1)</td>
<td>points C to F (line ( f_1 ))</td>
<td>( f_{i}(C(A)) = C(C) = 0.8(0.2 - \ln 2)(C(A) - 1) )</td>
</tr>
<tr>
<td>BAR(_2)</td>
<td>points A to B (( g_1 )), B to C (line), C to F (line), and F to A (line)</td>
<td>( g_{i}(C(A)) = C(C) = (0.2 - \ln 2)8C(A) - 34 - 4C(A) + \ln \left( \frac{5}{4 - 4C(A)} \right) - 0.1 )</td>
</tr>
<tr>
<td>BAR(_3)</td>
<td>points E to G (( h_1 )), G to D (( h_2 )), D to C (line), C to F (line), F to E (line)</td>
<td>( h_{i}(C(A)) = C(C) = 0.426\left( \frac{15 - 4C(A)}{16 - 16C(A)} \right) + \ln \left( \frac{5}{4 - 4C(A)} \right) - 0.1 )</td>
</tr>
<tr>
<td>BAR</td>
<td>points Z to W (( j_1 )), W to Y (( j_2 )), Y to H (line), H to F (line), F to Z (line)</td>
<td>( j_{i}(C(A)) = C(C) = 0.2(0.2 - \ln 2)12C(A) - 23 - 8C(A) + \ln \left( \frac{5}{3 - 8C(A)} \right) - 0.2 )</td>
</tr>
</tbody>
</table>

Figure 4. Shrink-Wrap-like algorithm superset for \( t_n - t_{n-1} = 2 \text{ min} \) and varying grid size.

Figure 5. Attainable region (AR) for steady-state networks.

Figure 4 shows increasingly accurate approximations of a superset to the BAR for species A (\( x \)-axis) and species C (\( y \)-axis), ranging from a 100 \( \times \) 100 grid size to a 10 000 \( \times \) 10 000 grid size. The identified BAR superset is only slightly smaller than the steady-state AR found in the work of Manousiouthakis et al.\(^1\) (shown in Figure 5) at the high end of the steady-state AR, but significantly smaller at the lower end of
the y-axis intersection of the AR (0.1851 for BAR, versus 0.0 for steady-state AR). Fundamentally, the reason for this discrepancy is the causality restriction that has been imposed on the batch network; the point (0.0, 0.0) cannot be part of the BAR because no batch reactor trajectory with a feed in BAR can have this point as an exit. Similarly, the point (0.0, 0.1) cannot be part of BAR for the same reason. Repeated application of this argument within the aforementioned Shrink-Wrap-like algorithm leads to the lower boundary of the BAR superset being the line that goes through the network feed point (1.0, 0.0), a batch reactor exit that lies on the y-axis, and its corresponding feed. Solving the batch reactor equations in Appendix A for a reaction time of 2 min and an outlet concentration of A of 0.0 mol/L gives an inlet concentration of A of 0.0625 or \( \frac{1}{16} \) mol/L. Also using Appendix A, a batch reactor that feeds at (0.0625, \( C(C) \)) will exit at (0, \( C(C) + 0.01157 \)). The value of \( C(C) \) for which the network feed point, aforementioned reactor inlet, and reactor outlet points are all on the same line is:

\[
\frac{\tilde{C}(C)}{0.0625 - 1} = \frac{\tilde{C}(C) + 0.01157}{-1} \quad \text{\( \tilde{C}(C) = 15(0.01157) = 0.1736 \)}
\]

Thus, the batch reactor exit is:

\[
C(C) = 0.01157 = 16(0.01157) = 0.1851
\]

This is consistent with the Shrink-Wrap-like BAR code, which identifies the lowest concentration attainable on the C(C) axis as point H (0.0, 0.1851).

The BAR superset identified in Figure 4 is, in fact, the BAR itself. As discussed in detail in Appendix C, if the point of BAR, with the lowest value of C(A) is considered to be a batch reactor feed point, then this batch reactor’s exit is on the y-axis. Mixing this exit with the network feed results in a point with C(A) = 0.0625 that is below the point in BAR, with the same value of C(A). Repeated application of this batch reaction/mixing process is shown in Appendix C to reach the lower boundary of the previously identified BAR superset. A similar procedure is outlined in Appendix D for the upper boundary of the superset, thus establishing that the superset given in Figure 4 is the actual BAR.

These results suggest that if species C were an unwanted byproduct of this reaction set, choosing any network of batch reactors with a reaction/holding time of 2.0 min would give unacceptable results; a network of steady-state reactors (e.g., a CSTR) could yield practically no generation of C and, thus, would be superior to a network of batch reactors (except when the reaction/holding time becomes very, very small.)

**Breadth-First BAR Construction Algorithm Results for the Trambouze Example**

A “breadth-first” algorithm can be used to identify the BAR, for any given n; we show the case for n = 3 as depicted in Figure 6 and an enlargement of the top of the same region as Figure 7 to illustrate the convergence of the algorithm with an increasing number of grid points.

This result matches very well with the analytical calculation of BAR, that is given in Appendix B.

**Conclusions**

We have outlined a method for automatically identifying the attainable region for batch reactor networks for a given overall network time, for any number of components, and for any isothermal kinetic model. We have shown that a batch reactor model can be incorporated into the IDEAS framework, which is proven to identify the globally optimal network of batch reactors.

We have demonstrated that a Shrink-Wrap-like algorithm can be used to identify increasingly accurate approximations of a superset of the batch attainable region (BAR). We have also outlined a novel breadth-first algorithm that can be used to identify the batch reactor network BAR. We have applied both algorithms to a case study with Trambouze kinetics, and we have demonstrated that the breadth-first algorithm accurately identifies BAR for (n = 3), whereas the Shrink-Wrap-like algorithm accurately identifies not only a superset of BAR but the actual BAR itself. Iterative procedures that demonstrate how the BAR boundary, as identified by the Shrink-Wrap-like algorithm, can be reached with causal (forward moving in time) batch reaction and mixing operations are discussed in detail.

**Appendix A: Solution of Batch Reactor Model for Trambouze Kinetics**

The considered Trambouze kinetics \((k_1 = 0.025, k_2 = 0.2, k_3 = 0.4)\) satisfy the relation \((k_2^2 - 4k_1k_3)^{1/2} = 0\). Therefore, any batch reactor trajectory satisfies the following relations:

\[
C(A) = \frac{(2 - k_2\tau)\tilde{C}(A) - \alpha k_2\tau}{2k_3\tau C(A) + 2 + k_3\tau}
\]

\[
C(C) = \tilde{C}(C) + \alpha[2\ln(1 + \alpha k_3\tau + k_3\tau\tilde{C}(A)) - k_2]\]
where \( r \) is the reaction time of unit, \( \alpha \) is defined as:

\[
\alpha = \frac{k_2}{2k_3} = \frac{1}{4}
\]

\( \tilde{C}(A) \) and \( C(A) \) respectively represent the inlet and outlet concentrations of species A, and \( \tilde{C}(C) \) and \( C(C) \) respectively represent the inlet and outlet concentrations of species C.

For \( r = 2.0 \):

\[
C(A) = \frac{\tilde{C}(A) - 0.0625}{C(A) + 1.5} \rightarrow \tilde{C}(A) = \frac{1 + 24C(C)}{16 - 16C(C)}
\]

and:

\[
C(C) = 0.5 \ln (0.8\tilde{C}(A) + 1.2) - 0.1
\]

Appendix B: Analytical Generation of BAR0, BAR1, BAR2, and BAR3

BAR0 consists solely of the feed point: \( \{ (1, 0) \} = \{ (C_0(A), C_0(C)) \} \).

BAR1 is the convex hull (which, in this case, is a line) between two points: the feed and the exit of a batch reactor with \( r = 2.0 \) and with the feed point as its inlet. Based on the formulas given previously, the latter point is \( (C_0(A), C_0(C)) = (0.375, \ln(\sqrt{2}) - 0.1) \). We will call this point C. BAR1 can be defined as

\[
BAR_1 = \{ (C_1(C), C_1(C)) \}
\]

where:

\[
C_1(C) = f_1(C_1(A))
\]

\[
C_1(A) \in [0.375, 1.0]
\]

and:

\[
f_1(C_1(A)) \equiv 0.8(0.2 - \ln 2)(C_1(A) - 1)
\]

BAR2 is the convex hull of all batch reactor trajectories starting from points in BAR1 and BAR1. It can be defined as follows:

\[
C_2(A) = \frac{C_1(A) - 0.0625}{C_1(A) + 1.5} \rightarrow f_2(C_1(A)) \rightarrow C_2(A) \in [0.1667, 0.375]
\]

\[
C_2(C) = C_1(C) + 0.5 \ln (0.8C_1(A) + 1.2) - 0.1 = 0.8(0.2 - \ln 2)(C_1(A) - 1) + 0.5 \ln (0.8C_1(A) + 1.2) - 0.1 = f_3(C_1(A))
\]

The convexification of this region will be a line from one endpoint of the BAR1 line to either another endpoint of the new set, or a point intermediate on that new set, depending on whether the new set has an inflection point in the domain:

\[
C_2(C) = (0.2 - \ln 2) \left( \frac{8C_2(A) - 3}{4 - 4C_2(A)} \right) + 0.5 \ln \left( \frac{5}{4 - 4C_2(A)} \right) - 0.1 \equiv g_1(C_2(A))
\]

This result shows that such a point exists; we will call this point \((0.2486, 0.3204)\) point A. The function \( g_1 \) is concave on the domain; therefore, BAR2 is the area formed by the line from the feed point to point A, the part of the batch reactor trajectory from point A to point B \((\ell_{16}, 0.5(\ln 3) - 0.2)\), the line from point B to point C \((\ell_{16}, 0.5(\ln 2) - 0.1)\), and the line from point C to the feed point (see Figure B1).

BAR3 for this problem is comprised of all points attainable from batch reactor trajectories that start on the boundary of BAR2. We only need to consider extreme points, because the Trambouze reaction vector is dependent only on the concentration of species A; any reactor inlet point on a given vertical line that satisfies the relation:

\[
\tilde{C}(C) \leq C(C) \leq \tilde{C}(C)
\]

will be mapped to a point \((C(A), C(C))\) that lies between the points \((C(A), \tilde{C}(C))\) and \((C(A), \tilde{C}(C))\). Therefore one must only consider the reactor trajectory that extends the region the furthest for each point on the \( C(A) \) axis. Three separate segments need to be mapped (i.e., travel along the batch reactor trajectories for 2 min) to create BAR3: line AF, curve AB, and line BC. Line CF does not need to be mapped again, because its map was calculated previously and has already been considered in BAR2. The convex hull of the region that is defined by these
three mappings and BAR2 will give BAR3. Mapping line AF gives:

\[ C_3(C) = 0.4264 \left( \frac{15 - 40C_2(A)}{16 - 16C_2(A)} \right) + 0.5 \ln \left( \frac{5}{4 - 4C_2(A)} \right) - 0.1 \equiv h_1(C_3(A)), \quad C_3(A) \in [0.1064, 0.375] \]

Mapping curve AB gives

\[ C_3(C) = \frac{(12C_3(A) - 2)(0.2 - \ln 2)}{3 - 8C_3(A)} + \ln \frac{5}{3 - 8C_3(A)} - 0.2 \equiv h_2(C_3(A)), \quad C_3(A) \in [0.0625, 0.1664] \]

Mapping line BC gives

\[ C_3(C) = 2.4 \left( \frac{\ln \left( \frac{2}{3} \right) + 0.2}{1 + 24C_3(A)} \right) + 0.5 \ln \left( \frac{5}{4 - 4C_2(A)} \right) + 0.9(\ln 3) - 0.4(\ln 2) - 0.38 \equiv h_3(C_3(A)), \quad C_3(A) \in [0.0625, 0.1667] \]

A plot of these three mappings is given as Figure B2.

To form BAR3, we must make the convex hull of these new points with the points of BAR2. We first determine if \( h_1 \) has a tangent that intersects the feed point:

\[ h_1(C_3(A)) - 0 = \frac{dh_1(C_3(A))}{dC_3(A)} \cdot (C_3(A) - 1) \]

\[ C_3(A) = 0.2158 \in [0.1064, 0.375] \rightarrow -0.4264 \left( \frac{\ln \left( \frac{5}{4 - 4C_3(A)} \right)}{15/8} - 5C_3(A) \right) + 0.8C_3(A) + 0.5325 = [1 - C_3(A)] \ln \left( \frac{5}{4 - 4C_3(A)} \right) \]

We see that \( h_1 \) does, in fact, intersect the feed point and is concave, so there will be a convexification line from the feed point to point E (0.2158, 0.3495). Because \( h_2 \) is also concave and its derivative is equal to the derivative of \( h_1 \) at their intersection, there will be no further convexifications at the top of the region. The function \( h_3 \) is concave and the value of \( C_3(C) \) for \( h_3 \) at point B is greater than that of line CD at the \( C_3(A) \) of point B; therefore, there is a convexification line from point C to point D (0.0625, 0.393147) to finish BAR3. This completes the analytical generation of BAR3; it is the region formed by the line from the feed to point E on \( h_1 \), the function \( h_1 \) from point E to point G (0.1064, 0.3882), the function \( h_2 \) from point G to point D, the line from point D to point C, and line CF (previously defined). The final graph of BAR3 appears almost identical to the result from the breadth-first algorithm (Figure 6) and will not be reproduced.

**Appendix C: Analytical Generation of the Bottom of BAR**

The point in BAR3 with the lowest value of \( C(A) \) (0.0625, 0.3931). This point maps to the point (0, 0.4047). The equation for the line from this point to the feed point is \( C(C) = 0.4047 - (1 - C(A)) \). The point on this new line at \( C(A) = 0.0625 \) is \( C(C) = 0.3794 \). Mapping this new point back to the axis gives the point (0, 0.3910). This procedure can be repeated iteratively as follows; define \( C_i(C) \) as the \( i \)th \( C(A) \) value in the sequence and \( \Delta C(A) \) as the increase gained by mapping, where \( C_0(C) = 0.3931 \) and \( \Delta C(C) = 0.01157 \). The first three points in the sequence are then:

\[ C_1(C) = \left[ \frac{15}{16} \right] C_0(C) + \Delta C(A) \]

\[ C_2(C) = \left[ \frac{15}{16} \right] C_1(C) + \left[ \frac{15}{16} \right] \Delta C(A) \]

\[ C_3(C) = \left[ \frac{15}{16} \right] C_2(C) + \left[ \frac{15}{16} \right] \Delta C(A) \]

and \( C_o(C) \), in the limit as \( n \to \infty \), can be expressed as:

\[ \lim_{n \to \infty} C_n(C) = C_0(C) \lim_{n \to \infty} \left[ \frac{15}{16} \right] + \left[ \frac{15}{16} \right] \Delta C(A) \lim_{n \to \infty} \sum_{k=0}^{n} \left[ \frac{15}{16} \right] = 15\Delta C(A) = 0.1736 \]

This point maps forward along a reactor trajectory for 2.0 min to point H (0.1851), which is also the lowest point on the \( C(C) \) axis obtained through the Shrink-Wrap-like algorithm.

**Appendix D: Analytical Generation of the Top of BAR**

As shown in the construction of BAR2 and BAR3, at high values of \( C(A) \) (\( C(A) > 0.375 \)), the upper BAR boundary is a straight line that is tangent to the curve generated when each point of the upper BAR\( _n-1 \) boundary (which, again, is a straight line for large values of \( C(A) \)) is considered a feed to a batch reactor with a reaction time of \( \tau = 2.0 \) min. BAR2 has, as its upper boundary, the line AF, which is tangent to the curve CAB (at point A), which is generated when the points of the upper boundary of BAR1 (line CF) are considered as feeds to batch reactors with \( \tau = 2.0 \). Similarly, BAR3 has, as its upper boundary, the line EF, which is tangent (at point E) to the curve \( h_1(C_3(A)) \). This curve consists of points that are outlets of batch reactors that have, as feeds, the points on the upper boundary of BAR2 (line AF). This process will stop expanding BAR upward (as \( i \to \infty \)) only when the line ZF (that is, the upper boundary of BAR, line AF) is tangent at point Z to the curve that results from mapping the line ZF through the batch reactor map with \( \tau = 2.0 \). To identify this point Z (\( Z = (C_2(A), C_2(A)) \), the following conditions must hold:

1. All points on the line ZF satisfy:

\[ \frac{C_i(C)}{C_i(A) - 1} = \frac{C_2(A) - 1}{C_2(A) - 1} \]

2. The line ZF maps to the following curve:

![Figure B2](image-url)
\[\begin{align*}
C(C) &= \left(\frac{40C(A) - 15}{16 - 16C(A)}\right)C_Z(C) + \frac{0.5 \ln \left(\frac{5}{4 - 4C(A)}\right)}{0.1 + j_1(C(A))} \\
\text{(3) At } C(A) = C_Z(A), C(C) &= C_Z(C) \text{ and } dC(C)/dC(A) \text{ is given as:} \\
\frac{dC(C)}{dC(A)} &= \frac{C_Z(C)}{C_Z(A) - 1} \\
\text{where:} \\
C_Z(C) &= \frac{8(1 - C_Z(A))^2}{25 - 16(1 - C_Z(A))^2} \\
\text{Taking the derivative of the function } j_1 \text{ from condition 2 above and evaluating the derivative at } C_Z(A) \text{ yields:} \\
\frac{dj_1(C(A))}{dC(A)} \bigg|_{C(A)=C_Z(A)} &= \\
&= \frac{0.5(C_Z(A))^2 - C_Z(A) - \frac{25}{16}C_Z(C) + 0.5}{(1 - C_Z(A))^3} \\
\text{This point must be on the map of line } ZF \text{ (condition 2), giving the coordinates of point } Z, (C_Z(A), C_Z(C)) = (0.1579, 0.4156). \text{ Because, by definition, line } ZF \text{ and its map have the same slope at point } Z, \text{ there is no need to convexify the top of the BAR near point } Z. \text{ Next, we define point } W (0.05752, 0.4568) \text{ as the point to which point } Z \text{ maps after a batch reaction time of } 2.0 \text{ min; the portion of the mapping of line } ZF \text{ between point } Z \text{ and point } W \text{ will now be called curve } ZW, \text{ as defined by the function } j_1. \text{ The second derivative of } j_1, \text{ with respect to } C(A), \text{ is negative between point } Z \text{ and point } W, \text{ so the curve } ZW \text{ is concave between point } Z \text{ and point } W. \text{ We can now map curve } ZW, \text{ through the batch reactor map with } \tau = 2.0, \text{ until it reaches the } C(A) \text{ axis to complete the top of BAR:} \\
C(C) &= \left(\frac{30C(A) - 5}{6 - 16C(A)}\right)C_Z(C) + \frac{0.5 \ln \left(\frac{5}{3 - 8C(A)}\right)}{0.2 + j_2(C(A))} \\
\text{The map hits the } C(A) \text{ axis at } C(C) = 0.4666. \text{ We will call this point } Y (Y = (0.0, 0.4666)) \text{ and refer to the portion of the map of } ZW \text{ for } \tau = 2.0 \text{ min that remains in the physical region (i.e., } C(A) \geq 0) \text{ as curve } WY. \text{ A plot of line } ZF, \text{ curve } ZW, \text{ and curve } WY \text{ is given as Figure D1. The derivatives of curve } ZW \text{ at point } W \text{ and curve } WY \text{ at point } W \text{ are equal:} \\
\frac{dj_1(C(A))}{dC(A)} \bigg|_{C(A)=C_W(A)} &= \frac{4(6C_W(A) - 1)}{(1 + 4C_W(A))(7 - 12C_W(A))} = \\
&= \frac{dj_2(C(A))}{dC(A)} \bigg|_{C(A)=C_W(A)} \\
\text{In addition, curve } WY \text{ is concave between point } Y \text{ and point } W, \text{ so no convexification line is necessary to fill in the region; the top of the BAR is the line from the feed point } F \text{ to point } Z, \text{ the function } j_1 \text{ from point } Z \text{ to point } W, \text{ and the function } j_2 \text{ from point } W \text{ to point } Y. \text{ This turns out to be approximated extremely well by the Shrink-Wrap-like algorithm, which, in this case, has successfully identified the BAR itself.} \\
\text{Literature Cited} \\
(15) \text{ Mehta, V.; Kokossis, A. Non-Isothermal Synthesis of Homogeneous and Multiphase Reactor Networks. } AIChE J. 2000, 46, 2256. \\
\end{align*}\]